Brief Papers

Parallel Algorithms for Finding a Near-Maximum Independent Set of a Circle Graph

YOSHIYASU TAKEFUJI, LI-LIN CHEN, KUO-CHUN LEE, AND JOHN HUFFMAN

Abstract—**A parallel algorithm for finding a near-maximum independent set in a circle graph is presented. An independent set in a graph is a set of vertices, no two of which are adjacent. A maximum independent set is an independent set whose cardinality is the largest among all independent sets of a graph. The algorithm is modified for predicting the secondary structure in ribonucleic acids (RNA). The proposed system, composed of an n neural network array (where n is the number of edges in the circle graph or the number of possible base pairs) not only generates a near-maximum independent set but also predicts the secondary structure of ribonucleic acids within several hundred iteration steps. Our simulator discovered several solutions which are more stable structures, in a sequence of 359 bases from the potato spindle tuber viroid (PSTV), than the formerly proposed structures. The simulator was tested in solving other problems.**

I. Introduction

THIS paper introduces a parallel algorithm for finding a near-maximum independent set of a circle graph within several hundred iteration steps. The circle graph is very suited for computing the secondary structure of ribonucleic acids (RNA). Nonintersected edges in the circle graph provide information on the base pairs for the folding of the chains like the beads in a necklace. To generate the stable RNA structure, we want to maximize the number of nonintersected edges or base pairs. The proposed parallel algorithm is modified and used for predicting the RNA secondary structure.

Nucleic acids are the means by which information about the structure and function of a living organism is stored and passed on to the next generation. Nucleic acids are composed of only two types of molecules: deoxyribonucleic acids (DNA) and ribonucleic acids. The organic bases of RNA include two compounds (Cytosine and Uracil) and two molecules (Adenine and Guanine). The primary structure is determined by the sequence of those bases (C, U, A, and G). The secondary structure is determined by the folding of the chains into a two-dimensional shape. The folding of the chains into a three-dimensional shape is called the tertiary structure. Predicting the primary, secondary, and tertiary structures is extremely important, not only for fighting against viruses and genetic problems, but also for enhancing the biotechnology. Tinoco's RNA structure stability model is used to compare our simulation results to the existing structures proposed by other investigators. The structure stability computation can show us how good our algorithm is.

The proposed parallel algorithm not only generates a near-maximum independent set of a circle graph but also predicts the secondary structure of ribonucleic acids. It requires n processing elements, where n is the number of edges in the circle graph or the number of possible base pairs. Our simulator, based on the proposed algorithm, discovered the new structures in a sequence of 38, 55, and 359 bases from the potato spindle tuber viroid (PSTV), two of which are more stable than the formerly proposed structures. The simulator was used to solve other problems to test our algorithm.

We believe this is the first parallel/distributed processing attempt to solve RNA secondary prediction problems. This paper presents a clear comparison between the conventional RNA folding algorithms, the backpropagation algorithm by Qian and Sejnowski or by Holley and Karplus, and our algorithm. Although the proposed algorithm is parallel computing, the simulator is currently running on sequential machines of an HP Apollo 3500 computer and a DEC 3100 computer under a UNIX operating system. The state of the system can usually converge to the near-optimum solution within about 500 iteration steps. We believe this paper presents a major breakthrough in not only computer science/engineering fields, to solve such NP-complete problems, but also in molecular biology fields.

The algorithm uses n processing elements where each processing element performs the following binary function: $V_i = 1$ if $U_i > 0$, 0 otherwise, where V_i and U_i are the output and input of the ith processing element. The processing element is called the McCulloch–Pitts binary neuron [1]. The first neural network for solving optimization problems was introduced by Hopfield and Tank [2].

Manuscript received January 19, 1990; revised April 6, 1990. This paper was supported in part by the National Science Foundation under Grant MIP-8902819.

Y. Takefuji, L.-L. Chen, and K.-C. Lee are with the Department of Electrical Engineering and Applied Physics, Case Western Reserve University, Cleveland, OH 44106.

J. Huffman is with the Department of Computer Engineering, Case Western Reserve University, Cleveland, OH 44106.

IEEE Log Number 9036590.

Takefuji and Lee successfully used the Hopfield neural network with McCulloch–Pitts binary neurons to solve the graph planarization problem [3], the tiling problem [4], and the sorting problem [5].

II. Finding a Near-Maximum Independent Set of a Circle Graph

An independent set in a graph is a set of vertices, no two of which are adjacent. A maximum independent set is an independent set whose cardinality is the largest among all independent sets of a graph. The problem of finding a maximum independent set for arbitrary graphs is NP-complete [6], [7]. Gavril developed a $\theta(n^3)$ time algorithm for finding a near-maximum independent set in a circle graph, where n is the number of edges in the circle graph [8]. Supowit proposed an $O(n^2)$ time algorithm in the circle graph [9]. Hsu gave an $O(m^4)$ time algorithm on the planar perfect graphs, where m is the number of vertices [10]. Choukhmane [11] and Burns [12] proposed an algorithm on cubic planar graphs. Masuda gave an $O(n \log n)$ time algorithm on the circle graphs [13]. Several parallel algorithms have been proposed by Karp [14], Luby [15], and Goldberg [16], where the computation time is $O((\log m)^4)$.

Consider the simple circle graph with 14 vertices and 7 edges as shown in Fig. 1(a). The adjacency graph can be obtained from the edge-intersection in the circle graph. For example, the edge "d" intersects with three edges: c, e, and f. The adjacency graph $G(V, E)$ of the circle graph in Fig. 1(a) is given by $V = \{a, b, c, d, e, f, g\}$ and $E = \{(a, b), (b, c), (b, e), (b, f), (c, d), (d, e), (d, f), (e, f), (f, g)\}$. A set of vertices $\{a, c, e, g\}$, as shown in Fig. 1(b), is the maximum independent set of this circle graph, which is equivalent to the maximal planar subgraph. It is given by $G(V, E)$ with $V = \{1, 2, \cdots, 14\}$ $E = \{(1, 13), (4, 7), (3, 11), (8, 10)\}$, where a set of edges $(1, 13), (4, 7), (3, 11), (8, 11)$ in Fig. 1(a) is equivalent to a set of vertices $(a, c, e, g\}$ in Fig. 1(b). In other words, to find the maximum independent set of the adjacency graph in the circle graph is equivalent to finding the maximum planar subgraph in the circle graph. In order to find the near-maximal planar subgraph in the circle graph with m vertices and n edges, n neurons (processing elements) are used in our algorithm. The output state of the ith neuron $V_i = 1$ means that the ith edge is not embedded in the circle graph. The state of $V_i = 0$ indicates that the ith edge is embedded in the circle graph. The motion equation of the ith McCulloch–Pitts binary neuron for $i = 1, \cdots, n$ is given by the following equation, where $d_{xy} = 1$ if the xth edge and the yth edge intersect each other in the circle graph, 0 otherwise.

$$\frac{dU_i}{dt} = A\left(\sum_{j=1}^{n} d_{ij}(1 - V_j)\left(\text{distance}(i)\right)^{-1}\right)(1 - V_i) - Bh\left(\sum_{j=1}^{n} d_{ij}(1 - V_j)\right)V_i. \quad (1)$$

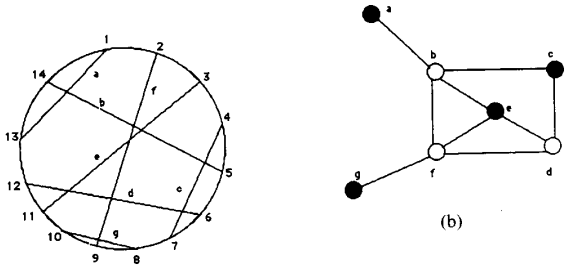

Fig. 1. (a) The circle graph with 14 vertices and 7 edges. (b) The maximum independent set.

Edge-intersection conditions between the ith and jth edges in the circle graph are given by: $\text{head}(i) < \text{head}(j) < \text{tail}(i) < \text{tail}(j)$ and $\text{head}(j) < \text{head}(i) < \text{tail}(j) < \text{tail}(i)$ where $\text{tail}(i)$ and $\text{head}(i)$ are two end vertices of the ith edge. Note that $\text{distance}(i)$ is given by $\text{distance}(i) = \min(|\text{head}(i) - \text{tail}(i)|, |n + \text{head}(i) - \text{tail}(i)|)$ where $\text{tail}(i) > \text{head}(i)$. The function $h(x)$ is 1 if $x = 0$, 0 otherwise.

The first term is the inhibitory force used to remove the edges which intersect with the ith edge in the circle graph. If the ith edge is removed from the circle graph, the first term will not be activated at all because the state of the ith neuron should be $V_i = 1$. In order to keep the ith edge in the circle graph, the first term should not have any edge-intersection violation. Whenever the ith edge has any edge-intersection violation, it will eventually be removed from the circle graph. The last term is the encouragement force used to embed the ith edge in the circle graph. If the ith edge is removed but does not intersect with any other edges, the last term will force the ith neuron to be $V_i = 0$. In other words, the ith edge is encouraged to exist in the circle graph.

The following procedure describes the proposed parallel algorithm.

0) Set $t = 0$.

1) The initial values of $U_i(0)$ for $i = 1, \cdots, n$ are set to small negative numbers or randomized.

2) Evaluate values of $V_i(t)$ for $i = 1, \cdots, n$ based on the binary function.

$$V_i(t) = \begin{cases} 1 & \text{if } U_i(t) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

3) Use the motion equation in (1) to compute $\Delta U_i(t)$.

$$\Delta U_i(t) = A\left(\sum_{j=1}^{n} d_{ij}(1 - V_j(t))\left(\text{distance}(i)\right)^{-1}\right) \cdot (1 - V_i(t)) - Bh\left(\sum_{j=1}^{n} d_{ij}(1 - V_j(t))\right)V_i(t).$$

4) Compute $U_i(t+1)$ on the basis of the first-order Euler method.

$$U_i(t+1) = U_i(t) + \Delta U_i(t)\Delta t \qquad \text{for } i = 1, \cdots, n.$$

5) Increment t by 1. If $\Delta U_i(t) = 0$ for $i = 1, \cdots, n$, then terminate this procedure or go to step 2).

One example which was investigated is the circle graph with 15 vertices and 27 edges. The simulation result was where the cardinality of the independent set is 13 out of 27 edges. The near-maximum planar subgraph contained 13 edges, which is equivalent to the solution. It took 28 iteration steps with $A = B = 1$ and the time unit step $\Delta t = 1$. A variety of circle graphs was investigated for verifying our motion equation to find the near-maximum independent set in the circle graphs. The simulation result shows the robustness of our algorithm.

III. Application to RNA Secondary Structure Prediction

Fresco used the first RNA secondary structure model for predicting the secondary structure in ribonucleic acids [17]. Two types of RNA folding algorithms have been reported: the "combinatorial" method, introduced by Pipas [18], and the "recursive" or dynamic programming method, introduced by Nussinov [19]. Both algorithms, including the latest method proposed by Zuker [20], are all based on sequential computation. Unfortunately, few parallel algorithms based on molecular thermodynamics models have been reported. Recently, Qian and Sejnowski [21] and Holley and Karplus [22] have reported a back-propagation algorithm using a three-layer feed-forward neural network for a protein secondary-structure prediction. Their method is based on the correlation between secondary structures and amino acid sequences. However, they have the following drawbacks over the conventional RNA folding algorithms, based on molecular thermodynamics models. 1) They need a teacher to force the network to learn the correlation between secondary structure and amino acid sequences. 2) The correlation models cannot provide an accurate prediction if a completely uncorrelated new datum is given, where the previously learned correlation information is useless. 3) Their feed-forward neural network requires a prohibitively long learning process to deal with a long sequence of bases for the RNA secondary structure prediction. Their methods cannot be applied to large-size prediction problems unless a prohibitively long learning time is permitted. 4) No theorem is given to determine the neural network architecture, including how many hidden layers and how many hidden neurons per hidden layer should be used.

Our algorithm requires neither a teacher nor a learning process. The proposed parallel algorithm, using n processors (where n is the number of possible base pairs), can yield the suboptimum solution within several hundred iteration steps. Our goal is to maximize the number of edges in the planar circle graph, where an edge represents possible base pairs, such as a G–C base pair or an A–U base pair. In the viroid structure prediction problem, Diener supports our goal of maximizing the number of base pairs [23].

The motion equation of (1) is used for predicting the secondary structure of ribonucleic acids with two modifications. One is that edge-intersection violation conditions must be updated. The total of six conditions to describe the edge-intersection in the circle graph are required: $\text{head}(i) < \text{head}(j) < \text{tail}(i) < \text{tail}(j)$, $\text{head}(j) < \text{head}(i) < \text{tail}(j) < \text{tail}(i)$, $\text{tail}(i) = \text{tail}(j)$, $\text{tail}(i) = \text{head}(j)$, $\text{head}(i) = \text{head}(j)$, and $\text{head}(i) = \text{tail}(j)$. The last four violation conditions are newly added to the first two conditions because a single base must not be involved in more than one base pair. The other modification is in the $\text{distance}(i)$ function of (1), where $\text{distance}(i)$ is given by $\text{distance}(i) = |\text{head}(i) - \text{tail}(i)|$.

A sequence of m bases is given to the simulator. It generates the circle graph with m vertices and n edges, where n is the number of possible base pairs, including G–C base pairs and A–U base pairs. The possible base pairs must also satisfy the hairpin-loop constraints, such as $|\text{head}(i) - \text{tail}(i)| > 3$. Because Tinoco [24] stated that it is sterically impossible to organize the hairpin loop with less than three bases, the circle graph is fed to the neural network simulator to find the near-maximum independent set.

Our simulator was tested by solving several secondary structure prediction problems in ribonucleic acids. In this paper, only one example is shown. A sequence of 38 bases from residues 1118–1155 of E. coli 16S rRNA, given by Stern [25], was used. Fig. 2(a) shows the secondary structure proposed by Stern [25] where the strength of the structure's stability is computed based on Tinoco's stability number [24]: 1) A–U pair, +1; 2) G–C pair, +2; 3) G–U pair, 0; 4) hairpin loops, −5 to −7; 5) interior loops, −4 to −7; 6) bulges, −2 to −6. For the details of Tinoco's stability model, see the paper [24]. The stability number of the secondary structure in Fig. 2(a) is +7. Fig. 2(b) shows the circle graph with 38 vertices and 151 edges. When $A = B = 1$ and $U_i(0) = -5$ for $i = 1, \cdots, 151$, the state of the system converged to the solution in the 104th iteration step. Fig. 2(c) shows the simulation result which contains 14 edges. The secondary structure of the simulation result is given in Fig. 2(d). The stability number of the structure is +11. It indicates that the simulator found a more stable structure than that of Stern's. Table I shows the simulation results where several sets of the coefficients were used.

Not shown here is a sequence of 55 bases from an R17 viral RNA [24] which was investigated. The circle graph has 55 vertices and 331 edges. When the following parameters were used: $A = 1$, $B = 0.01$, and $U_i(0) = -5$ for $i = 1, \cdots, 331$, the state of the system converged to the solution in the 161th iteration step. Our algorithm embedded 20 edges in the circle graph, where the structure stability is +7, while the stability of the structure proposed by Tinoco is +8. For predicting the secondary structure, several sets of coefficients were used.

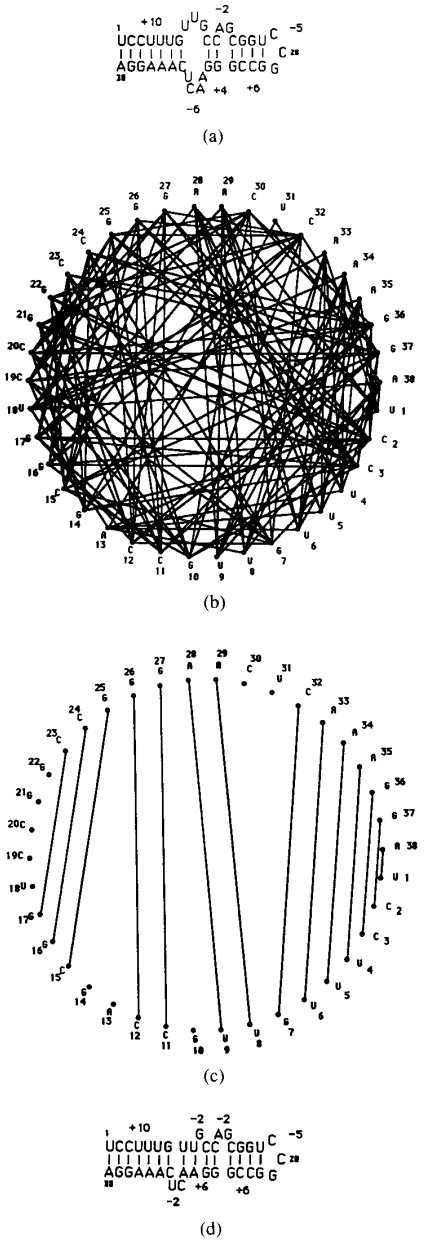

Fig. 2. (a) Secondary structure of 38 bases by Stern. (b) The circle graph with 38 vertices and 151 edges. (c) The maximum planar subgraph with 14 edges. (d) The simulation result of (c).

Finally, a sequence of 359 bases from the potato spindle tuber viroid (PSTV) was used to test our algorithm. Gross [26] proposed the secondary structure of the PSTV, where the stability number is +62. The circle graph had 359 vertices and 1017 edges and was generated where possible base pairs (i and j) were given by the following condition: $350 < i + j < 370$. The proposed condition drastically reduces the number of possible base pairs from more than 100 000 pairs to 1017 pairs. The state of the system converged to the solution in the 240th iteration step with $A = 1$, $B = 0.01$, and $U_i(0) = -5$ for $i = 1, \cdots, 1017$. The simulation result of the near-maximum planar subgraph had 359 vertices and 127 edges. In the secondary structure of the simulation result, the stability number is +50. Another simulation run was performed when $A = 1$, $B = 0.01$, and small negative random numbers were assigned to $U_i(0)$ for $i = 1, \cdots, 1017$. The secondary structure is composed of 359 vertices and 128 edges, where the stability number is +65. Sanger proposed another secondary structure of the PSTV, where the stability number is +64 [27]. A variety of the simulation runs were performed where several sets of the coefficients were used. The simulation result indicates that our simulator found the most stable structure of the PSTV. Our simulation result shows that, within about 500 iteration steps, the state of the system can converge to the solution in the PSTV secondary structure prediction problem.

TABLE I
THE SIMULATION RESULT: THE RELATION BETWEEN THE COEFFICIENTS, THE NUMBER OF ITERATION STEPS, AND THE NUMBER OF EMBEDDED EDGES

Coefficients		No. of iteration steps	No. of embedded edges
A	B		
1	0.2	24	14
1	0.3	34	14
1	0.5	54	14
1	1	104	14
1	2	204	14
4	0.05	80	13
2	0.05	34	12
4	0.5	23	12
2	0.5	84	11

IV. CONCLUSION

In this paper, we have shown the parallel algorithm for finding a near-maximum independent set and predicting the secondary structure of ribonucleic acids. The algorithm uses n processing elements, where n is the number of edges in the circle graph or the number of possible base pairs in ribonucleic acids (RNA). Our simulation result shows that the state of the system converges to the solution within several hundred iteration steps. The simulator discovered that the most stable structures in a sequence of 38 bases and a sequence of 359 bases from the PSTV were within 500 iterations.

REFERENCES

[1] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, p. 115, 1943.
[2] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141-152, 1985.
[3] Y. Takefuji and K. C. Lee, "A near-optimum parallel planarization algorithm," *Science*, vol. 245, pp. 1221-1223, Sept. 1989.

[4] ——, "A parallel algorithm for tiling problems," *IEEE Trans. Neural Networks*, vol. 1, no. 1, Mar. 1990.
[5] ——, "A super parallel sorting algorithm based on neural networks," *IEEE Trans. Circuits Syst.*, vol. 37, no. 7, 1990.
[6] R. M. Karp, *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972.
[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
[8] F. Gavril, "Algorithms on circular-arc graphs," *Networks*, vol. 4, pp. 357-369, 1974.
[9] K. J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Trans. Circuits Devices*, vol. 6, Jan. 1987.
[10] W. L. Hsu, "The coloring and maximum independent set problems on planar perfect graphs," *J. ACM*, vol. 35, no. 3, July 1988.
[11] E. Choukhmane and J. Franco, "An approximation algorithm for the maximum independent set problem in cubic planar graphs," *Networks*, vol. 16, 1986.
[12] J. E. Burns, "The maximum independent set problem for cubic planar graph," *Networks*, vol. 19, 1989.
[13] S. Masuda and K. Nakajima, "An optimal algorithm for finding a maximum independent set of a circular-arc graph," *SIAM J. Comput.*, vol. 17, no. 1, Feb. 1988.
[14] R. M. Karp and A. Wigderson, "A fast parallel algorithm for the maximum independent set problem," *J. ACM*, vol. 32, no. 4, Oct. 1985.
[15] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM J. Comput.*, vol. 15, no. 4, Nov. 1986.
[16] M. Goldberg and T. Spencer, "A new parallel algorithm for the maximal independent set problem," *SIAM J. Comput.*, vol. 18, no. 2, Apr. 1989.
[17] J. R. Fresco, B. M. Alberts, and P. Doty, "Some molecular details of the second structure of ribonucleic acid," *Nature*, vol. 188, no. 98, 1960.
[18] J. M. Pipas and J. E. McMahon, "Method for predicting RNA secondary structure," *Proc. Nat. Acad. Sci. USA*, vol. 72, no. 2017, 1975.
[19] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman, "Algorithm for loop matching," *SIAM J. Appl. Math.*, vol. 35, no. 68, 1978.
[20] M. Zuker, "On finding all suboptimal foldings of an RNA molecule," *Science*, vol. 244, no. 48, 1989.
[21] N. Qian and T. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *J. Molec. Biol.*, vol. 202, no. 865, 1988.
[22] L. H. Holley and M. Karplus, "Protein secondary structure prediction with a neural network," *Proc. Nat. Acad. Sci. USA*, vol. 86, 1989.
[23] T. O. Diener, *The Viroid*. New York: Plenum, 1987.
[24] I. Tinoco, O. C. Uhlenbeck, and M. D. Levine, "Estimation of secondary structure in ribonucleic acids," *Nature*, vol. 230, 1971.
[25] S. Stern, B. Weiser, and H. F. Noller, "Model for the three-dimensional folding of 16 S ribosomal RNA," *J. Molec. Biol.*, vol. 204, 1988.
[26] H. J. Gross *et al.*, "Nucleotide sequence and secondary structure of potato spindle tuber viroid," *Nature*, vol. 273, 1978.
[27] H. L. Sanger, *Minimal Infectious Agents*, B. W. J. Mahy and J. R. Pattison, Eds. London: Cambridge University Press, 1984.