IEEE Int. Workshop VLSI Design & Video Tech.
Suzhou, China, May 28~30, 2005

Using Stochastic Chaotic Simulated Annealing in the FPGA Segmented Channel Routing Problem

Lipo WANG [1,2] and Wensong ZHANG [1]

[1]School of Electrical and Electronic Engineering
Nanyang Technological University
Block S1, 50 Nanyang Avenue, Singapore 639798
[2]College of Information Engineering
Xiangtan University, Hunan, China

ABSTRACT

Field Programmable Gate Array (FPGA) segmented channel routing is a difficult combinatorial optimization problem. Neural networks have been successfully used to solve this problem. The Hopfield Neural Network (HNN) is easily stuck in local minimum but not global minimum, resulting in difficulties in finding the optional solution. This paper applies Stochastic Chaotic Simulated Annealing (SCSA) to the FPGA Segmented Channel Routing Problem (FSCRP) and compares the solutions of SCSA and the HNN.

1. INTRODUCTION

Since the first FPGA was implemented in the 1980's, it has been one of the fastest and most dynamic development techniques in electronics design. [1].

Figure 1 shows the architecture of such an FPGA [2][3][4], which consists of two rows of logic cells, and three horizontal tracks in the channel. For tracks 1, 2, there is one anti-fuse dividing each horizontal track into two horizontal segments. For track 3 there are two anti-fuses divide the horizontal track into three horizontal segments. These routing resources are present before programming. After functional and logic design, the final circuit design is used to program (blow) the cross-fuses and anti-fuses to provide a low resistance bi-directional connection between two segments-- vertical with horizontal and horizontal with horizontal [3].

Due to the resistance of the fuses, programming more fuses will add more delays [2][3]. Therefore, it is important to reduce the number of programmed fuses as far as possible. Each I/O pin requires only one essential cross-fuse, so we need only consider programmed anti-fuse to reduce the delays. Here, the number of programmed anti-fuses is called costs. Lower routing costs are better.

The connection of two I/O pins is called a net. Two nets cannot share the same horizontal segment but can share the separated horizontal segments in one track.

The aim of FSCRP is reducing routing costs to the minimal while ensuring no short circuit.

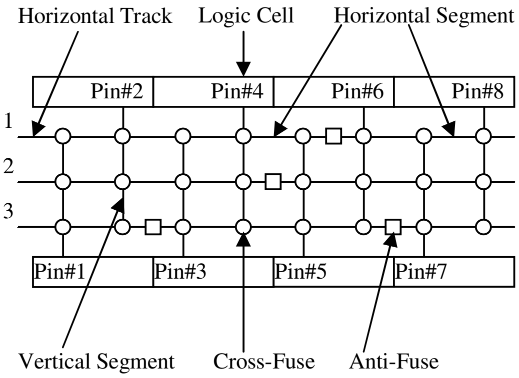

Fig.1. Row-based FPGA channel architecture

FSCRP is a NP-complete problem. Along with the increase in the number of nets and tracks, the computational complexity will increase exponentially [5][6][7].

For solving FSCRP, Funabiki etc. proposed the GNN algorithm which could get the better results than others methods [9].

But the major shortcoming of the HNN is that its energy function may be converged to a local minimum which could be far away from the global minimum. SCSA can overcome this shortcoming [10][11]. In this paper, we will try to solve the FSCRP by SCSA, based on the formulations given by Funabiki etc. [9].

2. GRADUAL NEURAL NETWORK ALGORITHM

In [9], the GNN steps of solving FSCRP are given by Funabiki etc. as follows.

The routing cost of net #n routing in track #m w_{nm} is the sum of the numbers of anti-fuse which needs to program.

Using routing costs, to compute the weighted coefficients B_{nm} and C_{nm} by:

$$nw_{nm} = w_{nm} - \min_{n}(w_{nm}),$$

$$B_{nm} = 1 + \left\lfloor \frac{B_0}{nw_{nm}} \right\rfloor, \text{ and } C_{nm} = 1 + \left\lfloor \frac{C_0}{nw_{nm}} \right\rfloor \quad (1)$$

The function $\lfloor x \rfloor$ returns the maximum integer which is smaller than or equals to x. In [9], setting $B_0 = 5, C_0 = 12$.

To check whether short circuit or not, defines:

$d_{n1.n2.m} = 1$, if different nets #n1 and #n2 share the same horizontal segment on track #m;

$d_{n1.n2.m} = 0$, if net #n1 and net #n2 are routed on track #m simultaneously but in separated horizontal segments. (2)

FSCRP is to minimize $\sum_{n=1}^{N} \sum_{m=1}^{M} w_{nm} x_{nm}$ (3)

when $\sum_{m=1}^{M} x_{nm} = 1$ (4)

and $\sum_{m=1}^{M} \sum_{m=1, n1 \neq n2}^{M} d_{n1.n2.m} x_{n1.m} x_{n2.m} = 0$ (5)

Equation (4) means each net must be routed in any track and be routed only once. Equation (5) ensures that there is no short circuit.

From equation (4)(5), can obtain the energy functions.

$$E_{A-term} = \frac{A}{2} \sum_{n=1}^{N} (\sum_{m=1}^{M} x_{nm} - 1)^2 \quad (6)$$

In [9], coefficient A is set to 1.

$$E_{B-term} = \frac{B}{2} \sum_{n1=1}^{N} \sum_{m=1}^{M} \sum_{n2=1, n2 \neq n1}^{N} d_{n1.n2.m} x_{n1.m} x_{n2.m} \quad (7)$$

And by motion function equation: $M = \frac{dy}{dt} = -\frac{\partial E}{\partial x}$ (8)

Can obtain the motion function of FSCRP:

$$M = M_{A-term} + M_{B-term} + M_{C-term} \quad (9)$$

where: $M_{A-term} = -A(\sum_{m=1}^{M} x_{nm} - 1)$ (10)

The B-term of motion function using the omega function:
if (iterative times t mod 12)<10,

$$M_{B-term} = -\sum_{n2=1, n2 \neq n1}^{N} B_{n2.m} d_{n1.n2.m} x_{n2.m} x_{n1.m}$$

Else, $M_{B-term} = -\sum_{n2=1, n2 \neq n1}^{N} B_{n2.m} d_{n1.n2.m} x_{n2.m}$ (11)

C-term is hill climbing term, trend to add one output if net #n has not routed in any track.

$$M_{C-term} = C_{nm} h(\sum_{m=1}^{M} x_{nm}) \quad (12)$$

where function h(x) returns: if x=0, h(x)=1; otherwise, h(x)=0.

Besides these methods, Funabiki etc. applied a Gradual Expansion Scheme, set a coefficient P, only when iterative times attain integer times of P=500, the neuron can be activated in batches.

In [9], if neuron input $y_i(t) > UTP = 63$ then neuron output $x_j(t+1) = 1$; and if $y_i(t) < LTP = 32$ then $x_j(t+1) = 0$; otherwise $x_i(t+1) = x_i(t)$. (13)

GNN algorithm steps are as follows.

Initialize neurons input $y_{nm,}$ output x_{nm}; compute Δy_{nm} by the sum of A,B,C terms' motion functions in batches; update neurons inputs by $y_{nm} = y_{nm} + \Delta y_{nm,}$ and if $y_{nm} < 0,$ then $y_{nm} = 0$; update x_{nm} by equation (13); terminate if satisfy termination checking equation (4) and (5), otherwise go to loop to compute Δy_{nm}. To set $t_{\max} = 3000$.

Funabiki etc. applied the above algorithm to solve FSCRP and obtained better results than others algorithms [9].

3. STOCHASTIC CHAOTIC SIMULATED ANNEALING ALGORITHM

Up to now, SCSA algorithm has been widely applied to various optimization problems and others problems with considerable success[10][12][13][14]. Different from HNN, the energy function of SCSA does not always decline monotonously [11]. Its general trend is from high energy level to low energy level, but often inversed from low energy level to high energy level temporary, at last inclining towards a stable low energy level. SCSA is formulated as equations (14) to (18).[10]

$$x_i(t) = \frac{1}{1 + e^{-y_i(t)/\varepsilon}}, \quad (14)$$

$$y_i(t+1) = ky_i(t) + \alpha(\sum_{j=1, j \neq i}^{n} w_{ij} x_j(t) + \theta_i) - z_i(t)(x_i(t) - I_0) + n(t), \quad (15)$$

$$z_i(t+1) = (1-\beta) z_i(t), \quad (16)$$

$$A[n(t+1)] = (1-\beta) A[n(t)], \quad (17)$$

and $\sum_{j=1, j \neq i}^{n} w_{ij} x_j + \theta_i = -\frac{\partial E}{\partial x_i}$ (18)

The variation direction of network condition of SCSA is not fixed. [15] and [16] give the detail specifications about it. Thereby, it is easy for SCSA to jump out of local minimum, arrive at deeper minimum or global minimum.

The steps of SCSA algorithm for FSCRP are as follows.

1) According to the anti-fuse locations and net sets, compute $d_{n1.n2.m}$ by equation (2); B_{nm}, C_{nm} by equation (1);
2) Randomly initialize y_{nm} by uniform distribution between -1 and +1, and initialize each $x_{nm} = 0$;
3) Compute motion function by equation (9) in turn;
4) By equation (8) and (18), we know that motion function equals the second term of equation (15), then update y_{nm} by equation (15) in turn;
5) Update x_{nm} by (14) in turn;
6) Terminal check: terminate if (4) and (5) are true or $t > t_{\max}$, otherwise go to loop to 3)

TABLE1 SCSA-GNN COMPARISON

Instance	# of tracks M	# of columns L	Unit segment length S	# of nets N	Average net length	SCSA			GNN		
						Solution times	Average cost	Lowest cost	Solution times	Average cost	Lowest cost
#1	16	128	8	60	18.97	97	48.23	43	5	48.2	44
#2					18.77	88	41.76	36	2	43.5	43
#3					17	100	32.01	29	100	38.26	34
#4					16.02	91	32.32	27	93	34.44	29
#5					15.7	100	30.27	25	100	33.17	28
#6	24	192	12	75	31.8	100	33.42	26	88	41.5	36
#7					31.03	59	33.84	28	11	40.55	37
#8					30.76	100	20.41	15	100	33.35	25
#9					30.51	65	31.92	26	3	38.33	37
#10					30.11	96	24.58	18	78	34.94	24

For comparing the result of SCSA and GNN, used C code to develop a simulator to solve 10 FSCRP instances by the two algorithms with the same energy functions, motion functions and coefficients. Get the results are showed in TABLE1.

The unit segment lengths S are the shortest segments, and will be added S to next four tracks. The start routing columns of nets are randomly generated, and the nets lengths are followed the gamma distribution [17]. Whether FSCRPs are difficult or easy to solve depend on the average net length, start columns of nets and gamma distribution parameters.

In SCSA, the selected parameters are as follows.

K=0.9, å =0.1, I_0 =0.001, á =0.005, â =0.1, z (0)=0.95, A[n(0)]=0.1 or 0

Each instance is solved 100 times with different random initialize numbers by SCSA and GNN individually. TABLE1 shows that the SCSA solutions are better than GNN in every instance.

Setting $t_{\max}$ =4000 when simulation. To run on PENTIUM 4-3.0GHz PC, SCSA will find one solution in most difficult instances with any initial inputs within 90 seconds. SCSA and GNN take the similar time with approximate iterative times. Usually SCSA finds solution and terminates the procedure with less iterative times, therefore SCSA is faster than GNN.

4. CONCLUSION

Solving FSCRP by two algorithms of GNN and SCSA has been compared in this paper. GNN can find better solutions than existing algorithms in 1999, however GNN only can get local minimum but not global minimum. SCSA can find deeper minimum or global minimum. In this paper, using the same energy function, motion function and coefficients, solving same instances by SCSA and GNN, the results of SCSA are better than HNN.

5. REFERENCES

[1] Stephen Brown, Jonathan Rose, FPGA and CPLD Architectures: A Tutorial, 1996 IEEE IEEE DESIGN & TEST OF COMPUTERS, PP.42-57

[2] Michael John Sebastian Smith, Application-Specific Integrated Circuits, Addison-Wesley Publishing Company, June 1997

[3] N. Sherwani, Algorithms for VLSI Physical Design Automation: Third Edition. Kluwer Academic Publishers,1999

[4] Y.-W. Chang, J.-M. Lin, and D. F. Wong, "Graph matching-based algorithms for FPGA segmentation design," Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD-98), pp. 34-39, Santa Clara, Nov. 1998.

[5] J. Greene, V. Roychowdhury, S. Kaptanoglu, and A. E. Gamal, "Segmented channel routing," in Proc. 27th Design Automat. Conf., 1990, pp. 567-572.

[6] A. E. Gamal, J. Greene, and V. Roychowdhury, "Segmented channel routing is nearly as efficient as channel routing (and just as hard)," in Proc. 13th Conf. Advanced Research VLSI, 1991, pp. 192-211.

[7] V. P. Roychowdhury, J. W. Greene, and A. E. Gamal, "Segmented channel routing," IEEE Trans. Computer-Aided Design, vol.12, pp. 79-95, Jan. 1993.

[8] JJ Hopfield, DW Tank, "Neural Computation of Decisions in Optimization Problems", Biological Cybernetics,Vol.52, pp.141-152, 1985.

[9] Nobuo Funabiki, M. Yoda, Junji Kitamichi, Seishi Nishikawa: "A gradual neural network approach for FPGA segmented channel routing problems." IEEE Transactions on Systems, Man, and Cybernetics, Part B 29(4): 481-489 (1999)

[10] Lipo Wang, Sa Li, "Channel Assignment for Mobile Communications Using Stochastic Chaotic Simulated Annealing", IWANN (1) 2001: 757-764

[11] Lipo Wang and Kate Smith, "On chaotic simulated annealing," IEEE Transactions on Neural Networks, vol.9, no.4, pp.716-718, July, 1998.

[12] Lipo Wang, Sa Li, Fuyu Tian, and Xiuju Fu, "A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing", IEEE Trans.

System, Man, Cybern, Part B - Cybernetics, vol.34, no.5, pp.2119-2125, October, 2004.
[13] Leipo Yan, Lipo Wang and Kim Hui Yap, "A noisy chaotic neural network approach to image de-noising," IEEE International Conference on Image Processing (ICIP 2004), Singapore, October, 2004.
[14] Lipo Wang and Haixiang Shi, "A noisy chaotic neural network approach to topological optimization of a communication network with reliability constraints," International Symposium on Neural Networks (ISNN 2004), Dalian, China, August, Lecture Notes in Computer Science, vol.3174, pp.230-235, 2004.
[15] Lipo Wang, "Oscillatory and chaotic dynamics in neural networks under varying operating conditions," IEEE Transactions on Neural Networks, vol. 7, no. 6, pp. 1382-1388, November, 1996.
[16] Lipo Wang, E.E. Pichler, and John Ross, "Oscillations and chaos in neural networks: an exactly solvable model," Proc. National Academy of Sciences (USA), vol. 87, pp. 9467 - 9471, 1990.
[17] M.Pedram, B.S. Nobandegani, and B.T.Preas, "Design and analysis of segmented routing channels for row-based FPGA' s," IEEE Trans. Computer-Aided Design, vol. 13, pp.1470-1479, Dec.1994.