Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **3**, May **2018**

A New Frog Leaping Algorithm Based on Simulated Annealing and Immunization Algorithm for Low-power Mapping in Network-on-chip

Lin Teng[1] and Hang Li[1,*]

[1]Software College
Shenyang Normal University
Shenyang, 110034 - China
ysl352720214@163.com;910675024@qq.com
*Corresponding author:910675024@qq.com

Received July, 2017; revised February, 2018

Abstract. *With the improvement of Network-on-chip integration, low-power mapping gradually becomes a hot research. In this paper, due to frog leaping algorithm searching with slow speed in the late of iteration and easily falling into local extremum, we propose a new frog leaping algorithm. Simulated annealing and immunization algorithm are introduced into frog leaping algorithm. Then we apply the new method into low-power Network-on-chip mapping. Finally, the experimental results show that proposed algorithm can further reduce the power consumption of communication.*
Keywords: Network-on-chip, low-power mapping, frog leaping algorithm, simulated annealing, immunization algorithm

1. **Introduction.** Network-on-Chip (NoC) [1] is an important development direction of System-on-Chip [2] (SoC). NoC transmits computer network technology into chip design. It solves the bottleneck problem in SoC from system architecture, which provides a new paradigm for complex SoC design. However, because the size of NoC circuit is very huge based on nanometer technology processing. So in system design constraints, how to reduce the power consumption of the system communication by mapping has become a research focus in the NoC design.

Chen [3] improved XBFT structure in power consumption by analyzing the topology routing hops to roughly determine the power consumption of routing topology. Zang [4] presented a novel shuffled frog-leaping algorithm with the adaptive chaos tactic and strategy of multi-neighborhood annealing used for solving the NoC mapping problem. Li [5] proposed an optimization scheme based on improved ant colony algorithm. Firstly, the parameters were for initialization operation. Secondly, tabu list was used to solve them, and the solutions were for local optimization of optimal solutions by using 2-opt algorithm. Collotta [6] presented a novel radio access control mechanism (RACM). The communication technologies based on the NoC concept had emerged with the aim of improving the scalability limitations of conventional NoC-based architectures. Among them, wireless NoCs (WiNoCs) used the radio medium for reducing the performance and energy penalties of long-range and multi-hop communications.

Frog leaping algorithm (FLA) was a kind of heuristic optimization algorithm based on group intelligent, which was proposed by Muzaffar[7]. FLA has some advantages with simple concept, less adjustment parameters, fast calculation speed, strong global optimization ability [8] etc,. At present, FLA has been applied into water resource distribution [9], traveling salesman problem [10], image processing [11] and workshop process arrangement. However, FLA's applications on NoC are little. So we use adjustment sequence idea to apply FLA into NoC low-power mapping, meanwhile, we improve the FLA by introducing simulated annealing and immunization algorithm. The experimental results show that, compared to the traditional FLA algorithm and simulated annealing FLA algorithm, new FLA algorithm has better search ability, which can further reduce the power consumption of communication. This paper is organized as follows. Section2 introduces the NoC mapping. We detailed explain the new FLA and apply it into network-on-chip mapping in section3. There is a conclusion in section4.

2. **Overview of Network-on-chip mapping.** In NoC mapping, objective function is the judgment index for mapping result [12-13]. Process of mapping is that it looks for a corresponding relationship between resource nodes and IP core. With this mapping relationship, the objective function can get the optimal value. In order to better describe the NoC mapping problem, the following two definitions are presented.

Definition 1. Using a characteristic pattern $A_{RCG}(R, P)$ as directed acyclic weighted graph. Each vertex $v_i \in V$ denotes an IP core, arc $e_{i,j} \in E$ denotes the communication relationship between v_i and v_j. $w_{i,j}$ is weight of $e_{i,j}$ that expresses the communication traffic between v_i and v_j.

Definition 2. $A_{RCG}(R, P)$ is as a directed graph. Each vertex $r_i \in R$ is a resource node in NoC. Directed $p_{i,j} \in P$ is the path from r_i to r_j. $e(r_i, r_j)$ is weight of $p_{i,j}$ that expresses the consumed energy transmitting one data from r_i to r_j.

Under the given characteristic pattern G_C and structure characteristic pattern G_R condition, $|G_C| \leq |G_R|$, NoC low-power mapping problem is to find a mapping function meeting:

$$\begin{cases} min \sum_{\forall c_i, c_j \in C} w_{i,j} e(\phi(c_i), \phi(c_j)) \\ \forall c_i \in C, \phi(c_i) \in R \\ \forall c_j \in C, \phi(c_j) \in R \end{cases} \tag{1}$$

The following is the explanation for power model. Unit bits data transmission between adjacent node resources consumes energy E_{bit}:

$$E_{bit} = E_{S_{bit}} + E_{B_{bit}} + E_{W_{bit}} + E_{L_{bit}}. \tag{2}$$

Where $E_{S_{bit}}$, $E_{B_{bit}}$ and $E_{W_{bit}}$ are consumed energy in exchange structure, buffering and internal wiring respectively. $E_{L_{bit}}$ is consumed energy in physical link. Under the existing technological level, size of resource nodes in the NoC has reached nanometer level, energy consumption in exchange structure, buffering and internal wiring is far less than that in the physical link.

$$E_{B_{bit}} + E_{W_{bit}} \prec E_{L_{bit}}. \tag{3}$$

So formula (3) can be written as:

$$E_{bit} = E_{S_{bit}} + E_{L_{bit}}. \tag{4}$$

Therefore, energy consumption of unit bits data transforming from t_i to t_j is:

$$E_{bit}^{t_i,t_j} = (h_{t_i,t_j} + 1)E_{S_{bit}} + h_{t_i,t_j} E_{L_{bit}}. \tag{5}$$

Where h_{t_i,t_j} is switch from resource node t_i to t_j denoted by Manhattan distance. Assuming that NoC 2D Mesh is in a x-y coordinate axis, so $h_{t_i,t_j} = |X_{t_i} - X_{t_j}| + |Y_{t_i} - Y_{t_j}|$. Size of NoC 2D Mesh is $N \cdot N$. Total energy consumption of information interaction in NoC is:

$$E_{NoC} = \sum_{i=1}^{N} \sum_{j=1}^{N} W_{t_i,t_j} \cdot E_{bit}^{t_i,t_j}. \tag{6}$$

Where W_{t_i,t_j} is communication traffic from resource node t_i to t_j. We can see that when the power consumption model is simplified, the direct optimization way for reducing the energy loss is to reduce the sum of the Manhattan distance. More specifical, it reduces the Manhattan distance with bigger communication weight between mapping resource nodes. After computing and comparing the power consumption under each mapping result, we can find the minimum energy consumption as formula (7),

$$minE_{NoC} = \sum_{i=1}^{N} \sum_{j=1}^{N} W_{t_i,t_j} \cdot E_{bit}^{t_i,t_j}. \tag{7}$$

3. **Low-power Mapping based on new Frog leaping algorithm.**

3.1. **Basic frog leaping algorithm.**

In a d-dimension target space, it randomly generates D frogs to constitute group $S = X_1, \cdots, X_D$. Position of $i - th$ fog denotes the solution $X_i = (x_{i1}, \cdots, x_{id})$. Then it calculates the fitness of each frog $f(X_i)$. It sorts the frogs from the best to the worst according to fitness value. These frogs will be divided into N groups $Y^1, \cdots, Y^N$ based on classifying criteria. Each group contains M frogs. And $D = N \times M$. Frog ranking first is allocated into first group. Frog ranking second is allocated into second group. Until frog ranking N is allocated into $N - th$ group. And frog ranking $N + 1$ is allocated into first group and so on. Then all the frogs will be allocated. So $Y^j = X_{j+N(l-1)} \in S, 1 \le l \le M, 1 \le j \le N$.

It starts to local search in each frog subgroup. Namely, in each current iteration, it needs to determine the optimal individual position X_b, the worst individual position X_w and global optimal individual position X_g. Then it only updates the X_w. Update strategy is the frog leaping length formula:

$$\Omega_i = rand() \cdot (X_b - X_w), -\Omega_{max} \le \Omega_i \le \Omega_{max}. \tag{8}$$

Frog individual position updating formula is:

$$X_w' = X_w + \Omega_i. \tag{9}$$

Where $rand()$ is the random number uniform distributing in [0,1]. Ω_{max} is the biggest frog leaping length. Executing (8) and (9), if the fitness of X_w' is superior to that in X_w, then X_w' replaces X_w. Otherwise, update strategy becomes the frog leaping length updating formula:

$$\Omega_i = rand() \cdot (X_g - X_w), -\Omega_{max} \le \Omega_i \le \Omega_{max}. \tag{10}$$

Frog individual position updating formula:

$$X_w' = X_w + \Omega_i. \tag{11}$$

Executing (10) and (11), if the fitness of X_w' is superior to that in X_w, then X_w' replaces X_w. Otherwise, it randomly generates a new individual position to replace the original X_w.

$$X_w' = rand() \cdot (O_{max} - O_{min}). \tag{12}$$

Where O_{max} and O_{min} denote the maximum and minimum search range respectively. Repeat the update operation until meet predetermined local iteration number in subgroups. When all local searches are finished, it exchanges global information and remixtures, sorts frogs. Then it repeats the above local search until satisfying the iteration stopping condition.

3.2. **New frog leaping algorithm.** We introduce immunization algorithm into the global information exchange stage of frog leaping algorithm. First, we select global optimal individual X_g as vaccine. Second, we randomly select dimension position information needed to inoculation to conduct vaccination according to inoculation probability for every frog. Finally, it accepts offspring produced by vaccination according to annealing mechanism. Namely, it makes immune selection for every frog according to annealing probability. Then the immunization operation is finished. This operation improves the quality of the candidate solution obviously, and as the annealing temperature drop, probability of accepting inferior solution gradually decreases too, which improves the efficiency and precision of the optimization calculation, enhances the global optimization ability.

Meanwhile, we introduce Gaussian mutation and chaos perturbation operation in local search stage. We make a comparison to individual fitness of current group with average fitness of current subgroup. When the individual fitness value is optimal, it will execute Gauss mutation. Then we use annealing probability to accept mutated individuals. Gauss mutation formula is:

$$m(x) = x \cdot (1 + N(0, 1)). \tag{13}$$

Where x is the current individual. $N(0, 1)$ denotes normal distribution random number with expected value 0 and standard?deviation 1. $m(x)$ is the mutated individual. Otherwise, it makes chaos perturbation for current individual, it uses annealing probability to accept mutated individuals too. This operation improves population diversity in the late of algorithm. It jumps out of local optimal solution and accelerates convergence speed. Finally, it can obtain the global optimal value. We adopt Tent chaotic mapping [14], its disturbance operation processes are as follows:

1. Randomly generate a d-dimension chaotic variable $X = (X_1, X_2, \cdots, X_l, \cdots, X_d)$, where $X_l \in [0, 1]$ and $l = (1, 2, \cdots, d)$.
2. Conduct Tent chaotic mapping for x, namely:

$$X_l' = \begin{cases} 2X_l, 0 \le X_l \le 0.5 \\ 2(1 - X_l), 0.5 \le X_l \le 1 \end{cases} \tag{14}$$

3. Map X_l' into original optimal space.

$$newX_l = min_l + (max_l - min_l) \cdot X_l'. \tag{15}$$

   Where $[min_l, max_l]$ is the domain $i - th$ dimension variable. So the new chaos perturbation is $newX = (newX_1, newX_2, \cdots, newX_l, \cdots, newX_d)$.
4. For the current individual Y (its fitness is less than current average fitness), the new individual after chaos perturbation is:

$$newY = (newY + Y)/2. \tag{16}$$

5. According to the annealing probability $min1, e^{-\Delta f/T_k} > rand()$ to accept the individual after the chaos disturbance. Where $\Delta f = f(newY) - f(Y)$, f is fitness function. T_k is annealing temperature

So we construct a new frog leaping algorithm based on simulated annealing and immunization algorithm (abbreviated as SAIFLA). Its detailed iteration processes are as follows:

1. Random initializing frog population and initial chaos mapping variable value and setting parameters in SAIFLA.
2. Calculating fitness value of each frog.
3. Making immunization operation for each frog. 1): finding current global optimal individual X_g as vaccine; 2): randomly selecting dimension position information needed to inoculation to conduct vaccination according to inoculation probability p_m for every frog, these dimension position information will be replaced by corresponding dimension position information in vaccine, then finishing immunization; 3): if $min1, e^{-\Delta f/T_k} > rand()$, then individual with immunization will replace individual without immunization. Otherwise keeping original individual.
4. It sorts the frogs from the best to the worst according to fitness value and divides them into n subgroups.
5. For each subgroup, repeat conducting the following steps. 1): update the optimal individual position X_b of current subgroup, global optimal individual position X_g according to the fitness, and determine the worst individual position X_w of current subgroup; 2): update X_w according to formula (8-12) and then compute current average fitness f'; 3): all the individuals whose fitness is superior to f' will receive Gaussian mutation based on (13), if $min1, e^{-\Delta f/T_k} > rand()$, then individual with Gaussian mutation will replace individual without Gaussian mutation. Otherwise keep original individual; 4): all the individuals whose fitness is worse than f' will receive chaos perturbation based on (14-16), if $min1, e^{-\Delta f/T_k} > rand()$, then individual with chaos perturbation will replace individual without chaos perturbation. Otherwise keep original individual.

In NoC mapping problem, position vector of each frog denotes one feasible solution for NoC mapping, namely it corresponds a detailed NoC mapping scheme. Assuming frog individual $U = (U^1, U^2, \cdots, U^d)$, then it can get corresponding NoC mapping scheme according to coding rule. For example, one NoC system has nine IP cores, frog individual $(1, 3, 4, 7, 9, 8, 2, 5, 6)$ denotes that it maps IP cores $(1, 3, 4, 7, 9, 8, 2, 5)$ into corresponding node $(1, 2, 3, 4, 5, 6, 7, 8, 9)$ respectively.

4. **Experiments and results analysis.** In this section, we select eight randomly task graphs and use frog leaping algorithm (FLA), simulated annealing frog leaping algorithm (SAFLA), a new method based on bat algorithm (BAT)[15] and our simulated annealing immunization frog leaping algorithm (SAIFLA) to make experiment. The experiment is conducted under MATLAB platform with XP system, Intel Q9500 2.82GHz CPU and 4GB memory. SAFLA only makes simulated annealing optimization for better individual. Parameter initial population number $P = 10d$, frogs population number $F = 10$, initial temperature $T_s = 100°C$, cooling temperature $T = 0.01°C$. We use FLA, SAFLA, BAT and SAIFLA to solve every communication task graph. With the different IP cores, we use the optimization results of SAFLA as standard to make normalization process. Figure1 is the performance comparison with the four methods. The optimization result with SAIFLA is obviously better than FLA, SAFLA and BAT, which reduces by 35.1%, 17.1% and 8.9% respectively. Since SAIFLA has optimal initial population, what's more, simulated annealing and immunization algorithm are introduced into this new method. This method effectively suppresses the precocious phenomenon and improves the precision of the algorithm. Meanwhile, the overall trend of four curves in figure1 shows that performance of SAIFLA will be towards to the best with the increase of NoC mapping size. Figure2 is the power consumption comparison curve using the four methods with the increase of IP core size. We use the optimization power consumption results of SAFLA as

standard to make normalization process. From figure2, we can know that power consumption with SAIFLA increases slowly. When IP cores is the biggest, SAIFLA is only 8.25 times than benchmark power consumption, however, power consumption of FLA, SAFLA and BAT is 13.85, 10.99 and 6.31 times than benchmark power consumption respectively.

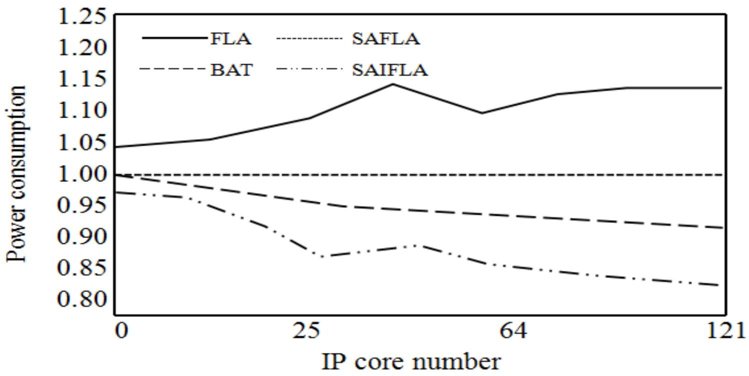

Figure 1. Power consumption comparison with three methods.

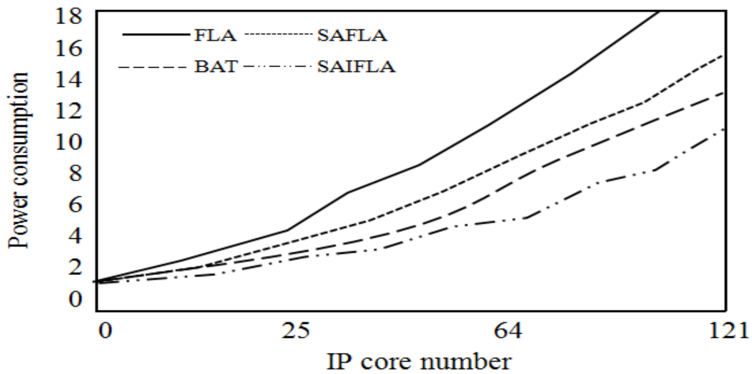

Figure 2. Power consumption comparison with increase of NoC size.

To further compare the optimization quality and running time, we adopt TGFF software to generate a certain size application feature graph, randomly generated communication bandwidth and delay constraint in the 3×3 and 6×6 structure for the different resource size. TGFF will randomly produce six PE application feature graphs denoted by a,b,c,d,e,f. The number of PE is from 9 to 64. Randomly generated communication bandwidth and delay constraint are as the input of FLA, SAFLA, BAT and SAIFLA. Figure3 shows the evolutionary generation. NEG denotes number of evolutionary generation. Note that, with the increasing of problem size, the NEG increases too. However, our method has the better competitiveness performance.

5. **Conclusions.** In this paper, we propose a new frog leaping algorithm based on simulated annealing and immunization algorithm to solve low-power mapping problem in Network-on-chip. The new method adopts the advantages of simulated annealing and immunization algorithm to improve the initial solutions for frog leaping algorithm. The experimental results show that the SAIFLA obviously can get optimal results and obtain global optimal solutions than FLA, BAT and SAFLA. And this new method greatly reduces the power consumption on NoC. In the future, we will study more advanced intelligence algorithms to perfect low-power mapping in Network-on-chip.

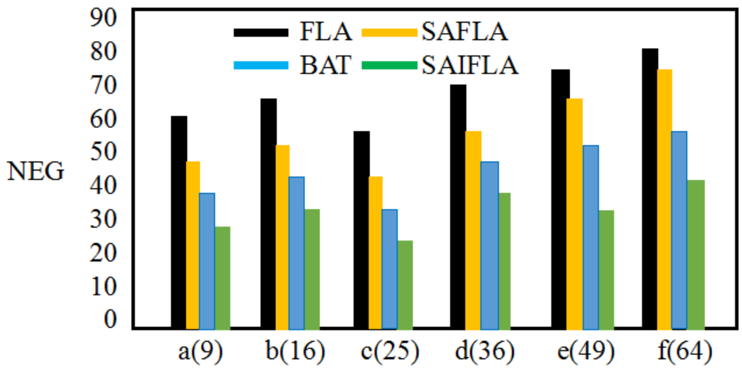

FIGURE 3. Evolutionary Generation Number.

Acknowledgment. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Kasapaki E, Schoeberl M, Srensen R B, et al. Argo: A Real-Time Network-on-Chip Architecture With an Efficient GALS Implementation, *IEEE Transactions on Very Large Scale Integration Systems*, vol. 24, no. 2, pp. 479-492, 2016.

[2] Dutt N, Jantsch A, Sarma S. Toward Smart Embedded Systems: A Self-aware System-on-Chip (SoC) Perspective, *Acm Transactions on Embedded Computing Systems*, vol. 15, no. 2, pp. 1-27, 2016.

[3] Chen H B, Li C. Improved XFBT: A Low-Power Topology of Network-on-Chip, *Advanced Materials Research*, no. 989-994, pp. 4865-4868, 2014.

[4] Zang M, Wang M, Zhou W, et al. Improved shuffled frog-leaping algorithm for low-power network-on-chip mapping, *Xian Dianzi Keji Daxue Xuebao/journal of Xidian University*, vol. 42, no. 1, pp. 118-123, 2015.

[5] Li D. Optimization on NoC Mapping Based on Improved Ant Colony Algorithm, *Applied Mechanics & Materials*, vol. 539, pp. 280-285, 2014.

[6] Collotta M, Palesi M, Mineo A, et al. An Efficient Radio Access Control Mechanism for Wireless Network-On-Chip Architectures, *Journal of Low Power Electronics & Applications*, vol. 5, no. 2, pp. 38-56, 2015.

[7] Eusuff M M, Lansey K E. Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm, *in Pro. of World Water and Environmental Resources Congress*, pp. 1-8, 2015.

[8] Jie L, Teng L, Yin S. An Improved Discrete Firefly Algorithm Used for Traveling Salesman Problem, *in Pro. of International Conference in Swarm Intelligence. Springer, Cham*, pp. 593-600, 2017.

[9] Molinos-Senante M, Mochol-Arce M, Sala-Garrido R. Estimating the environmental and resource costs of leakage in water distribution systems: A shadow price approach, *Science of the Total Environment*, vol. 568, pp. 180-188, 2016.

[10] Yin S, Liu J, Teng L. An Improved Artificial Bee Colony Algorithm for Staged Search, *TELKOMNIKA Telecommunication, Computing, Electronics and Control* , 14, no. 3, pp. 1099-1104, 2016.

[11] T. Lin, H. Li and S. Yin, Modified Pyramid Dual Tree Direction Filter-based Image De-noising via Curvature Scale and Non-local mean multi-Grade remnant multi-Grade Remnant Filter, *International Journal of Communication Systems*. 2017. DOI: 10.1002/dac.3486.

[12] Le Q, Yang G, Hung W N N, et al. A multiobjective scatter search algorithm for fault-tolerant NoC mapping optimisation, *International Journal of Electronics*, 2014, 101, no. 8, pp. 1056-1073.

[13] Yin S, Liu J, Teng L. An Improved Artificial Bee Colony Algorithm for Staged Search, *TELKOMNIKA Telecommunication, Computing, Electronics and Control.* , 14, no. 3, pp. 1099-1104, 2016.

[14] Liang S, Hao Q, Li J, et al. Chaotic optimization algorithm based on Tent map, *Control & Decision*, 2005, 20(2).

[15] Li J, Song G, Ma Y, et al. Bat Algorithm Based Low Power Mapping Methods for 3D Network-on-Chips, *in Pro. of National Conference of Theoretical Computer Science. Springer, Singapore*, pp. 277-295, , 2017.