

Improved Simulated Annealing Genetic Algorithm based Low power mapping for 3D NoC

Hanna He^{1,2}, Fang Fang¹ and Wei Wang^{1,2}

¹ Hefei University of Technology, 230009 HeFei, China

² School of Computer Science and Information Engineering, Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, 230009 HeFei, china

Abstract. Mapping of IP(Intellectual Property) cores onto NoC(Network-on-Chip) architectures is a key step in NoC-based designs. Energy is the key parameter to measure the designs. Therefore, we propose an Improved Simulated Annealing Genetic Alogrithm, abbreviated as ISAGA. The algorithm combines the parallelism of Genetic Algorithm(GA) and the local search ability of Simulated Annealing(SA). We improve the initial population selection of GA to get the lower power consumption mapping scheme. The experimental results show that compared with the GA, ISAGA has good convergence and can search the optimal solution quickly, which can effectively reduce the power consumption of the system. In the case of 124 IP cores, the average power consumption of the ISAGA is reduced by 32.0% compared with the GA.

1 Introduction

With the rapid development in nanometer and VLSI technology, integrating a large number of IP cores into a single chip will be a challenge for the System-on-Chip (SoC)¹. The emergence of the NoC architecture has completely solved the SoC problem from the architecture². However, with the rapid increase in the number of IP cores, 2D NoC faces a range of issues such as chip area, performance, bandwidth, and power consumption³. Therefore, the concept of 3D NoC was proposed⁴. Compared with 2D NoC, it has a smaller area and shorter delay, and has greatly improvement in performance and power consumption⁵.

The 3D NoC research is classified into three major categories: 3D NoC infrastructure, 3D NoC communication mechanism, and 3D NoC mapping optimization⁵. The NoC mapping algorithm is to assign the processing unit in the task graph to the NoC architecture resource node based on the given task characteristic diagram and the NoC architecture⁷. This article is based on the 3D-Mesh topology structure, and mainly researchs the mapping algorithm to the NoC architecture. Similar to the problem of task allocation, the IP core mapping problem is an NP-hard problem in which we explore the NoC design space to find the Pareto optimal mapping sequence⁸. At present, many researches have been carried out at home and abroad, such as Ascia G proposed a genetic algorithm to solve the multi-objective optimization problem of Noc mapping⁷. Chang proposed an improved tabu search algorithm to continuously search for the best mapping scheme by local search and elite reorganization¹⁰. Li proposed a power consumption model for 3D-mesh, implemented a NoC

mapping for communication energy consumption using ant colony algorithm based on the 2D NoC research method¹¹. Kartikeya Bhardwaj proposed C3Map and ARPSO Based Mapping Algorithms for Energy-Efficient 3D NoC Architectures¹². PK Hamedani proposed ILP based static thermal sensing mapping algorithm¹³. Xiao-Hang Wang proposed an energy efficient Run-time incremental mapping algorithm using TSV technology¹⁴. F Ge and G Feng proposed a thermal-aware mapping algorithm for 3D mesh network-on-chip architecture¹⁵. However, these mapping algorithms still have deficiencies and need to be further improved.

In this paper, we propose an Improved Simulated Annealing Genetic Algorithm(ISAGA) based low power mapping for 3D NoC architectures. Our main contributions are as follows:

- We obtain a better mapping scheme by improving the initial population. The improvement of the initial population can effectively improve the quality of the population, so that the algorithm can search for the optimal individual (the optimal mapping scheme) faster and more accurately.
- adding simulated annealing at the mutation operator stage. The addition of the simulated annealing algorithm makes the algorithm less likely to fall into the local optimum, and strengthens the global optimization ability of the algorithm.

Rest of the paper is organized in many sections. We describe the NoC mapping problem and give the power consumption model in Section 2. We introduce the principle of GA and ISAGA, and give the specific steps in Section 3. We use some relevant experiments to prove the point is effective in Section 4. Finally, we conclude our proposed work in Section 5.

2 3D NoC mapping model

2.1 Mapping Description

3D NoC mapping is to allocate IP cores to each resource node in NoC according to a certain mapping algorithm, so that the entire NoC has the smallest amount of traffic, and finally achieves the purpose of reducing power consumption based on the known NoC architecture and IP core traffic. The mapping process is shown in Figure 1. The subtasks correspond to the IP cores in the NoC architecture.

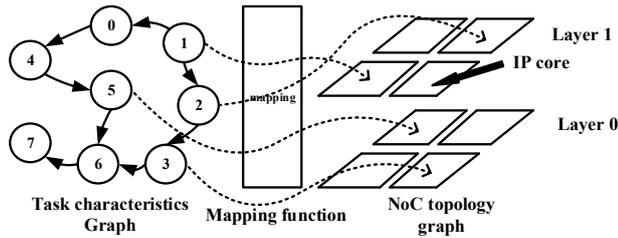


Figure 1. Mapping of task nodes to NoC architecture.

Definition 1 defines a weighted directed acyclic task feature graph $MCG = \langle V, T \rangle$, The vertex V_i represents a task, and the directed arc $t_{i,j} \in T$ represents the communication between the tasks t_i and t_j , their weights $\omega_{i,j}$ are the amount of communication data between them. If there is no communication relationship between t_i and t_j , then Equation (1) gives a matrix of communication relationships between all tasks $W = \{\omega_{i,j} | 0 \leq i \leq n-1, 0 \leq j \leq n-1\}$.

$$W = \begin{bmatrix} 0 & \omega_{0,1} & \cdots & \omega_{0,n-1} \\ \omega_{1,0} & 0 & \cdots & \omega_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n-1,0} & \omega_{n-1,1} & \cdots & 0 \end{bmatrix} \quad (1)$$

Definition 2 defines a NoC topology structure diagram $NTD = \langle E, P \rangle$. A vertex $e_i \in E$ represents a resource node and $p_{i,j} \in P$ represents a communication path between p_i and p_j . The weights $f_{i,j}$ of edges p_i and p_j are the communications volume between them.

Definition 3 For 3D-mesh type NoC, M_{dij} represents the jump distance(Manhattan distance) from e_i to e_j , is expressed as Equation (2).

$$M_{dij} = |x_i - x_j| + |y_i - y_j| + |z_i - z_j| \quad (2)$$

Therefore, the traffic $f_{i,j}$ of the resource nodes e_i to e_j can be expressed by Equation (3).

$$f_{i,j} = \omega_{i,j} \times M_{dij} \quad (3)$$

2.2 Power Consumption Model

The NoC system power consumption is mainly composed of the energy consumed by the processing node during the operation and processing and the energy consumed by the network topology in the data transmission¹⁷. The energy consumed by the network topology consists of two parts: the energy consumption of the routing node and the energy consumption of the communication path.. Therefore, in order to minimize the power consumption,

the power consumption model proposed in¹⁸ is adopted in this paper. The energy consumption for transmitting a basic length of 1_{bit} data at NoC can be expressed by Equation (4).

$$E_{bit} = E_{S_{bit}} + E_{B_{bit}} + E_{W_{bit}} + E_{L_{bit}} \quad (4)$$

Among them, $E_{S_{bit}}$, $E_{B_{bit}}$, $E_{W_{bit}}$, $E_{L_{bit}}$ represents the energy consumed in crossbars, buffers, internal interconnects, and adjacent routing nodes, respectively. Because $E_{B_{bit}}$ and $E_{W_{bit}}$ are very small actually, they are usually negligible, so change the formula (4) to

$$E_{bit} = E_{S_{bit}} + E_{L_{bit}} \quad (5)$$

Thus, the energy consumed by the 1_{bit} data from node e_i to node e_j is calculated as

$$E_{bit}^{D_{ij}} = (D_{ij} + 1) \times E_{S_{bit}} + D_{ij} \times E_{L_{bit}} \quad (6)$$

Among them, D_{ij} represents the number of nodes that have passed. Under the premise of adopting the shortest path route, $D_{ij} = M_{dij}$, thus, (6) can be expressed as

$$E_{bit}^{D_{ij}} = (M_{dij} + 1) \times E_{S_{bit}} + M_{dij} \times E_{L_{bit}} \quad (7)$$

The overall NoC system communication power is

$$E = \sum_{i,j} [(M_{d_{ij}} + 1) \times E_{S_{bit}} + M_{d_{ij}} \times E_{L_{bit}}] \times \omega_{ij} \quad i \in V, j \in V \quad (8)$$

By formula (3) available,

$$E = \sum_{i,j} f_{ij} \times (E_{S_{bit}} + E_{L_{bit}}) + \sum_{i,j} \omega_{ij} \times E_{S_{bit}} \quad i \in V, j \in V \quad (9)$$

In equation (9), all constants except $\sum_{i,j} f_{ij}$ are constants, so the objective function of low-power mapping can be expressed as

$$\min \sum_{i,j} f_{ij} \quad i \in V, j \in V \quad (10)$$

$$s.t. \text{map}(e_i) = v_i \quad \forall e_i \in E, \exists v_i \in V$$

In summary, we design a good mapping algorithm to reduce the amount of traffic and ultimately reduce the NoC communication power consumption.

3 Genetic Algorithms and Improvements

GA is a computational model based on Darwin's evolution theory and Mendel's genetic theory. It is a heuristic search method to search for optimal solutions by simulating the natural evolution process¹⁹.

As we all know, the IP core mapping problem belongs to the NP-hard problem. The coverage of GA is large, which is conducive to global optimization. At the same time, the algorithm is easy to implement parallelism. It has unique advantages in solving performance and computing time and is very suitable for solving IP core mapping problems.

This paper considers that the GA is too random and the initial population quality is low. Therefore, we propose a method to improve the initial population effectively, and considering the deficiency of the GA in

local optimization, we add the SA at the mutation operator stage so that the progeny could jump out of the local optimum and tend to be the global optimum. In the following section, we call it ISAGA. The entire flow of the algorithm is shown in Figure 2 (the dotted box in the figure is which we proposed in this paper).

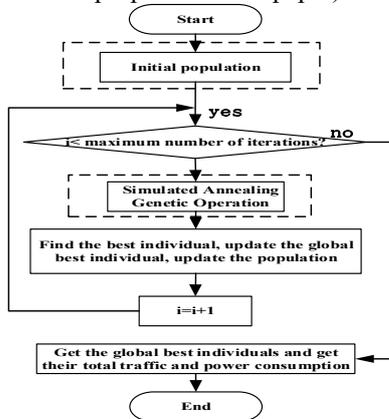


Figure 2. Simulated annealing algorithm flow chart based on improved initial population

3.1. Coding scheme

This paper adopts real number coding. The vector is used to represent the assignment of the task graph to the IP core. Assume vector $X = (x_1, x_2, \dots, x_n)$, where n is the number of IP cores and x_i is the IP core number. All IP cores form a chromosome, representing a mapping scheme.

A classic task graph MWD, as shown in Figure 3, in which IP core numbers 0 to 11 are mapped to $2 \times 2 \times 3$ 3D NoC, and each processing unit of NoC is represented by y_0 to y_{11} . The structure is shown in Figure 4.

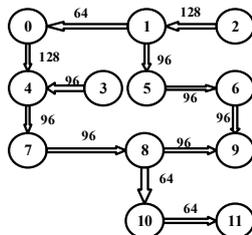


Figure 3. Classic task graph MWD

The vector $X=(0, 4, 1, 3, 5, 6, 2, 7, 11, 9, 10, 8)$ represents the IP core (0, 4, 1, 3, 5, 6, 2, 7, 11, 9, 10, 8) are mapped to 3D NoC's (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11) processing units. The result is shown in Figure 5.

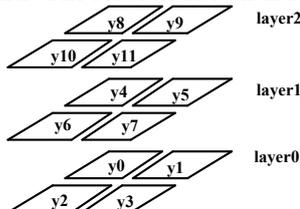


Figure 4. $2 \times 2 \times 3$ 3D NoC Structure

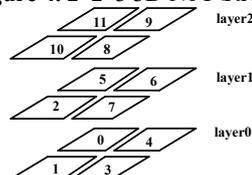


Figure 5. Mapping results

3.2 Initial population improvement

The initial population selection of the GA is random, so the population quality is low. We improved the initial population selection method, so that the best population can be obtained in tens of generations or a hundred generations, which increases the population's iteration speed and reduces the data traffic greatly, reducing the power consumption ultimately.

In20, a greedy algorithm is proposed to construct the initial population, so that the initial population quality is improved and the power consumption is reduced. However, when the initial population is generated, the algorithm requires a large amount of calculations for each individual. When determining the position where each individual is mapped to 3D NoC, it is necessary to calculate the fitness value of all unmapped individuals to all available positions, and all the individuals need to perform this operation, so the initial population running efficiency is not high.

We use a depth-first approach to traverse the task graph to determine where each individual is mapped to the 3D NoC, without calculating the fitness value of all unmapped individuals to all available positions. It is only necessary to calculate the Manhattan distance between it and its neighbors. Our method greatly reduces the amount of calculation and effectively improves the efficiency of the algorithm. The specific steps of the algorithm are as follows:

- Construct two arrays to store the node access flags of task graph MCG and topology graph NTD (nodes are numbered from 0).
- Starting from the vertex 0 of the MCG, access and mark it as visited, then place the task node at the 0th position of the NTD, and mark it as visited.
- Visit the neighbor of the vertex 0 of the MCG, if not visited, access and mark it as visited. Then traverse NTD, if a location is not visited and it has the shortest Manhattan distance to the previous visited vertex (manhattan distance is stored in the distance matrix), place the vertices of the MCG at this position, mark it as visited, place the position number in the array pop, and perform recursively.
- If the neighbor of vertex 0 has been visited, returned to the previous vertex. Then check whether the vertex's neighbor is visited. If there are any neighbors that have not been visited, continue to visit, if all have been visited, return to the previous vertex and continue the same operation.
- When all the vertices of the MCG have been visited, traverse all positions of the NTD, place the last node of the MCG in the last unvisited location and mark it as visited. The final result is stored in pop.
- Swap any two locations in the array pop randomly, variation generates new individuals, uses this method to generate the remaining individuals, and stores all the individuals in the Group array.
- Select the individual with the largest fitness value (where the minimum value of the objective

function $f_{i,j}$) from the Group array as the initial population by roulette selection.

3.3 Simulated Annealing Genetic Algorithm

The Simulated Annealing (SA) was proposed by N. Metropolis²¹ in 1953. It is a stochastic optimization algorithm based on the Monte-Carlo iterative solution strategy. It can jump out of the local optimal solution probabilistically and eventually tends to global optimality. It overcomes the poor local search ability and precocity of genetic algorithm. Combining it with GA will give full play to GA's global search ability and SA's local search ability.

Based on this consideration, we added SA at the mutation operator stage of the GA to ensure that the generated progeny do not fall into a local optimum, so that the population jumps out of the local optimum and eventually reaches a global optimum. The core idea of the algorithm: Let new individual fitness be f , the threshold of change is f' . When $f > f'$, accept new individuals, otherwise, accept new individuals with a certain probability $P = \exp((f - f')/T)$. where T is the temperature. The algorithm is described as follows:

- Initialization: initial temperature $T=T_0$, $i=0$.
- When $T > T_{final}$, do the following steps, otherwise, return to the optimal solution.
- If $i < =$ number of cross-operations, do the following steps, otherwise, return to the optimal solution.
- Select n pairs of individuals from the group as the parent, for each parent, do the following:
 - The parent P_1, P_2 cross operation to generate children S_1, S_2
 - If $f_{s1} > f_{p1}, f_{s2} > f_{p2}$, then use s_1, s_2 to replace p_1, p_2 for the next genetic operation, Otherwise, hold p_1, p_2 with the probability of $\exp((f_{s1} - f_{p1})/T)$.
- Drop temperature in a certain way, $T=T_i+1, i=i+1$.
- Return step3.

Where T_0, T_{final} indicate the initial temperature and the termination temperature, respectively.

4 Equations and mathematics Experimental results and analysis

The source code of the paper is written in C++ language, and the programming tool is Visual Studio 2012. It runs in the windows environment. In order to verify the feasibility of the algorithm, we use three classic task graphs for comparison experiments. And in order to reflect the universality of the algorithm, we also use a random task graph generated by the TGFF22 to perform a comparison experiment.

4.1 Parameter Design

This paper is a mapping algorithm based on GA. There are many parameters that will affect the performance of the algorithm. We can improve the efficiency of the

optimal solution by controlling of these parameters. Specific parameters are designed as shown in Table 1.

Table 1. Simulated annealing genetic algorithm parameters.

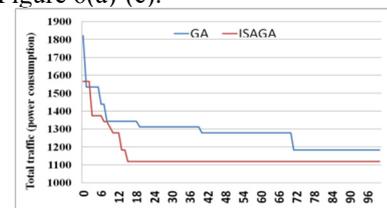
Parameter	Value
Population size	30
Hybridization rate	0.8
Mutation rate	0.01
Number of iterations	500
Initial temperature	1000
Termination temperature	0.01

4.2 Comparison and Analysis of Experimental Results

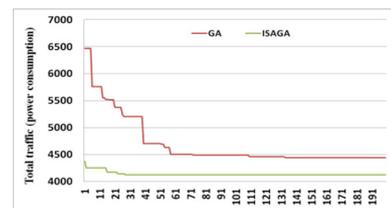
The three classic task graphs used in this experiment are MWD, VOPD and DVOPD. They are 12, 16, and 32 nodes, respectively.

4.2.1 Comparison of Convergence Rates Based on Classical APCG

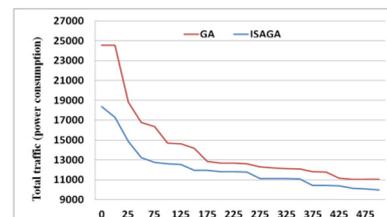
In this section, we compare the convergence speeds of the classic task graphs MWD, VOPD, and DVOPD. The abscissa represents the number of iterations, and the ordinate represents the total amount of communications. The smaller the total amount of communications, the smaller the system power consumption. The comparison of GA and ISAGA for MWD, VOPD and DVOPD is shown in Figure 6(a)-(c).



(a). MWD



(b). VOPD



(c). DVOPD

Figure 6. Comparison of Convergence Rates of Two Algorithms for Classic Task Graph

Table 2. Comparison and Analysis of Experimental Results

Method	MWD			VOPD			DVOPD		
	avg	max	min	avg	max	min	avg	max	min
Traditional	1134	1183	1119	4397.7	4865	4251	11823.3	12889	11062
Improved	1119	1119	1119	4125	4125	4125	10124.4	10412	9752

According to Figure6(a)-(c), with the number of iterations increases, the total traffic decrease gradually, the ISAGA has a significantly lower total communication volume than the GA. Due to the ISAGA has a higher quality in the initial solution, it will have a better solution in fewer iterations. The ISAGA has greater advantages in terms of convergence speed, and with the increase in the number of IP cores and traffic, this advantage will be even more pronounced.

4.2.2 Power Comparison Based on Classic APCG

Due to the randomness of the GA, for different APCG, we use GA and ISAGA to solve the average value of 30 times respectively, as shown in Table 2.

As shown in Table 2, analyzing the results of MWD, the average power consumption of the ISAGA is reduced by 1.3% compared to the GA, the maximum power consumption is reduced by 5.4%, and the minimum power consumption is flat. Although it has been reduced, the reduction is relatively small. Due to the small number of nodes, the GA and the ISAGA can find a better solution faster in a certain number of iterations.

Analyzing the VOPD results, compared with the GA, the ISAGA reduces the average power consumption by 6.2%, reduces the maximum power consumption by 15.2%, and reduces the minimum power consumption by 3.0%. In terms of maximum power consumption, the ISAGA has a significant reduction in power consumption compared to the GA, because the population of GA is generated randomly, the population quality is relatively low. After using the ISAGA, the initial population quality has been significantly improved compared to random.

Analyzing the DVOPD results, compared with the GA, the ISAGA reduces the average power consumption by 14.4%, the maximum power consumption by 19.2%, and the minimum power consumption by 11.8%. With the increase in the number of IP cores, it is clear that the ISAGA has a significant reduction in power consumption in all aspects. The ISAGA accelerates the convergence speed by improving the initial population quality. On the other hand, due to the global optimization ability of SA, the algorithm avoids falling into the local optimal solution. The combination makes the ISAGA have better power consumption than the GA.

4.2.3 Comparison of Power Consumption Based on Random Task Graph

TGFF was used to generate random task graphs with IP core numbers of 24, 45, 60, 80, 98, and 124. The task

graph attributes are shown in Table 3. For these random task graphs, two algorithms were used to solve the average of 20 times. The experimental results are presented as percentages. The experimental comparison results are shown in Table 4.

Table 3. Random task graph attributes.

Number	Number of IP cores	Task directed arc
1	24	33
2	45	67
3	60	90
4	80	108
5	98	156
6	124	174

Table 4. Reduction rate in power consumption.

IP cores	Reduced power consumption rate (%)					
	24	45	60	80	98	142
max	4.40	14.80	25.20	26.70	25.70	35.30
avg	2.70	12.90	20.70	23.40	25.00	32.00
min	-0.19	11.10	18.70	22.30	23.80	28.00

Comparing ISAGA and GA, we can see that when the number of IP cores is small, the reduction in power consumption is small, less than 10%. Because the number of nodes is too small, it is trapped in a local optimum, due to the number of nodes is too small, it is trapped in a local optimum, so the minimum power consumption is reduced by a negative growth. However, with the increase in the number of IP cores, the average power consumption reduction ratio has reached 32%, and the reduction ratio of the maximum power consumption has reached 35% in 124 IP cores, which is a very significant performance improvement. From the overall trend of the data in the table, with the increase of IP cores, this reduction will gradually increase.

4.2.4 Comparison with other improved genetic algorithms

Literature20 proposed an improved genetic algorithm based on greedy algorithm (Improved Genetic Alogrithm), refered to as IGA, which uses greedy algorithm to generate initial population to reduce power consumption. The algorithm proposed in this paper(Improved Simulated Annealing Genetic Alogrithm) is simply referred to as the ISAGA. In the following, the two improved genetic algorithms are compared by the classical task graph and the random task graph.

1) Classic task graph power comparison

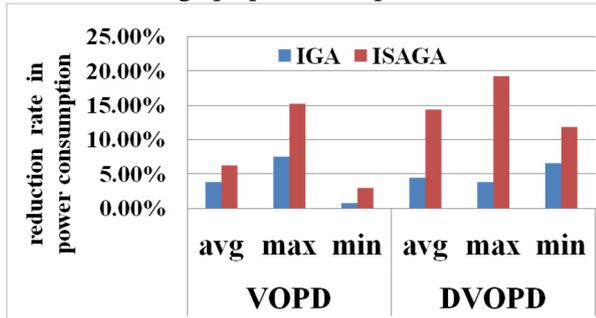


Figure 7. Two Algorithms for Classic APCG Power Reduction Comparison Chart

As shown in Figure 7, analyzing the VOPD comparison results, ISAGA reduces more power consumption than IGA. From the experimental results, it can be seen that the average power consumption is reduced by 38.9%, the maximum power consumption is reduced by 50.3%, and the minimum power consumption is reduced by 74.3%.

Analyzing of DVOPD comparison results, ISAGA compared to IGA power reduction is more obvious. From the experimental results, it can be seen that the average power consumption is reduced by 68.8%, the maximum power consumption is reduced by 80.1%, and the minimum power consumption is increased by 44.6%. Due to the relatively large number of DVOPD tasks, the initial population improved by ISAGA is not easy to fall into local optimum, and because of the addition of simulated annealing algorithm, the algorithm can jump out of local optimum and tend to be globally optimal.

From the above experimental results and analysis, we can see that when the number of IP cores is large, the advantage of ISAGA is more obvious.

2) Comparison of random task graph power consumption

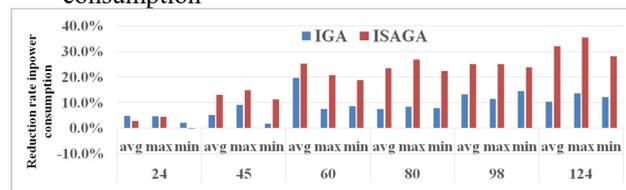


Figure 8. Two algorithms for random APCG power reduction comparison chart

As shown in Figure 8, in terms of average power consumption, when the number of IP cores is small, the IGA decreases slightly compared with ISAGA, because in the case of fewer tasks, the algorithm is prone to fall into a local optimum, however, with the increase in the number of IP cores, from the overall trend, the reduction in power consumption of ISAGA is gradually higher than that of IGA. In 124 IP cores, the power consumption reduction ratio of the ISAGA relative to the IGA is increased by 67.8%.

In terms of maximum power consumption, when the number of IP cores is 45, the power consumption reduction ratio of the ISAGA relative to the IGA is increased by 38.5%. When the number of IP cores is 80,

the power consumption reduction ratio of the ISAGA relative to the IGA is increased by 68.9%. When the number of IP cores is 124, the power consumption reduction ratio of the ISAGA relative to the IGA is increased by 61.4%.

In terms of minimum power consumption, when the number of IP cores is 45, the power reduction ratio of the ISAGA algorithm relative to the IGA is increased by 84.7%. When the number of IP cores is 80, the power consumption reduction ratio of the ISAGA relative to the IGA is improved. 65.0%. When the number of IP cores is 124, the reduction in the power consumption of the ISAGA relative to the IGA is increased by 57.1%.

From the above experimental data, it can be seen that with the increase of the number of IP cores, the ISAGA has more significant advantages in reducing power consumption, and can reduce power consumption better than the IGA.

5 Conclusion and Outlook

5.1 Conclusion

For the 3D NoC low-power mapping problem, we propose a genetic algorithm that improves initial populations and combines simulated annealing. We effectively solve the problem of combining the optimization algorithm with the mapping. Experiments show that ISAGA has a good effect in solving 3D NoC low-power mapping. In the case of 124 IP cores, the maximum power consumption generated by ISAGA is reduced by 35.3% compared with GA.

5.2 Outlook

1) Fever is an important issue to be considered in the design of 3D NoC. With the deepening of research, we will consider the factors of fever and other factors in the research of mapping algorithm.

2) At present, the research on mapping algorithms for heterogeneous 3D NoC architecture is still in its infancy, and there is still much room for development. In future research, heterogeneous features can be fully utilized, and different mapping algorithms are used for different types of task communication diagrams to improve the mapping efficiency of 3D NoC and reduce the power consumption of the system.

References

- WR Davis, J Wilson, S Mick. *Demystifying 3D ICs: The pros and cons of going vertical*. IEEE Design & Test of Computers, **22**:498-510(2005).
- M Horowitz, R Ho, K Mai.. *The Future of Wires*. Proc of the IEEE, **89**: 490-504(1999)
- D K ZHANG, C HUANG, G Z SONG. *A survey of three dimensional network on chip*. Journal of Software, **27**:155-187(2016).
- K Lee, SJ Lee, D Kim, K Kim. *Networks-on-chip and Networks-in-Package for High-Performance*

- SoC Platforms*. Asian Solid-state Circuits Conference, 485-488(2005).
5. VF Pavlidis, EG Friedman. *3-D Topologies for networks-on-chip*. IEEE Trans on Very Large Scale Integration Systems, 1081-1090(2007).
 6. UY Ogras, J Hu, R Marculescu. *Key research problems in NoC design:a holistic perspective*. CODES+ISSS, 69-74(2005).
 7. SY Jin, YC Yao, MW Qin. *An NoC mapping algorithm based on communication links*. Application of Electronic Technique, **8**(2016).
 8. R Ricci, C Alfeld, J Lepreau. *A solver for the network testbed mapping problem*. Acm Sigcomm Computer Communication Review, **33**:65-81(2003).
 9. G Ascia, V Catania, M Palesi. *Multi-objective mapping for mesh-based NoC architectures*. CODES+ISSS, 182-187 (2004).
 10. Z W CHANG, X N XIE, N SANG, G Z XIONG. *An improved tabu search algorithm for network-on-chip mapping*. Journal of CAD and CG, **20**: 155-160(2008).
 11. D S LI, Q LIU. *Research on mapping 3D network on chip for communication energy-aware*. Semiconductor Technology, 504-507(2012).
 12. K Bhardwaj, PS Mane. *C3Map and ARPSO based mapping algorithms for energy-efficient regular 3-D NoC architectures*. VLSI-DAT, 1-4(2014).
 13. PK Hamedani, S Hessabi, H Sarbaziazad, NE Jerge. *Exploration of Temperature Constraints for Thermal Aware Mapping of 3D Networks on Chip*. PDP 2008, 499-506(2012).
 14. XH Wang, P Liu, M Yang. *Energy Efficient Run-Time Incremental Mapping for 3-D Networks-on-Chip*. Journal of Computer Science and Technology, **28**:54-71(2013).
 15. G Feng, F Ge, S Yu, N Wu. *A thermal-aware mapping algorithm for 3D Mesh Network-on-Chip architecture*. Proc of the IEEE 10th International Conference on ASIC, 1-4(2013).
 16. F Ge, G Feng, S Yu, N Wu. *Power-and Thermal-aware Mapping for 3D Network-on-chip*. Information Technology Journal, **12**:7297-7304(2013).
 17. L Wang. *Research on on-chip network mapping optimization algorithm*. University of Electronic Science and Technology. (2010)
 18. L Tang, S Kumar. *A two-step genetic algorithm for mapping task graphs to a network on chip architecture*. Proc of Euromicro Symposium on Digital System Design, 180-187(2013).
 19. JH.Holland. *Adaptation in natural and artificial systems*. Quarterly Review of Biology, **6**:126-137(1975).
 20. H Z LIN, D K ZHANG, C HUANG. *Research on lower power mapping for three-dimensional network-on-chip based on improved genetic algorithm*. Computer Engineering & Application, **52**: 81-88(2016).
 21. M Steinbrunn, G Moerkotte, A Kemper. *Heuristic and Ran2 domized Optimization for the Join Ordering Problem*. The VLDB Journal, **6**: 8-17(1997).
 22. RP Dick, DL Rhodes, W Wolf. *TGFF: Task Graphs for Free*. Proc of the 1998 Conf on Hardware/Software Codesign, 97-101(1998).