

A novel mapping algorithm for three-dimensional network on chip based on quantum-behaved particle swarm optimization

Cui HUANG, Dakun ZHANG (✉), Guozhi SONG

School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2017

Abstract Mapping of three-dimensional network on chip is a key problem in the research of three-dimensional network on chip. The quality of the mapping algorithm used directly affects the communication efficiency between IP cores and plays an important role in the optimization of power consumption and throughput of the whole chip. In this paper, basic concepts and related work of three-dimensional network on chip are introduced. Quantum-behaved particle swarm optimization algorithm is applied to the mapping problem of three-dimensional network on chip for the first time. Simulation results show that the mapping algorithm based on quantum-behaved particle swarm algorithm has faster convergence speed with much better optimization performance compared with the mapping algorithm based on particle swarm algorithm. It also can effectively reduce the power consumption of mapping of three-dimensional network on chip.

Keywords three-dimensional network on chip, mapping algorithm, quantum-behaved particle swarm optimization algorithm, particle swarm optimization algorithm, low power consumption

1 Introduction

Increased computational complexity of communication terminals and more sophisticated devices have led to the higher level of integration of communication chips [1]. As the number of IP cores integrated in system on chip (SoC) has risen

dramatically recently, on-chip interconnection increasingly becomes the bottleneck of system performance due to the limitations of the traditional bus structure [2]. To solve this problem, Dally is the first person to propose a new method of communication between IP cores on chips using packet routing. He named this communication method as network on chip (NoC) [3,4]. Three-dimensional network on chip (3D NoC) is an effective solution to the problem of interconnection complexity of large-scale SoC. 3D NoC breaks the limitations on performance and size of two-dimensional Network on Chip (2D NoC) by using integrated circuit stacking technology. However, the application of stacking technology has led to the increased density of transistor, inevitably causing overheating problem of the chip [5,6]. Optimization of mapping and layout [7] can reduce performance overhead and increase versatility and has become an important means to solve the heat problem for 3D NoC [8–10]. Therefore, it is necessary to develop better mapping algorithms of 3D NoC. Quantum-behaved particle swarm optimization (QPSO) algorithm has many advantages such as global convergence, less control parameters, robustness, fast convergence speed, strong ability of optimization and so on. It is a meaningful attempt to apply QPSO to the study of mapping problem of 3D NoC.

2 Basic problems of mapping of 3D NoC

2.1 Basic concepts of mapping of 3D NoC

Mapping of 3D NoC is used to decide which IP in a given application characteristic graph (APCG) should be represented

by which node of topology architecture graph of 3D NoC to achieve the goal of minimizing power consumption or packet latency described as objective functions [11,12]. Mapping problem belongs to quadratic assignment problem and is an NP-hard problem [13]. To formulate this problem, two definitions regarding the application tasks and the target 3D NoC architecture first introduced in Ref. [14] will be presented here.

Definition 1 An application characterization graph (APCG) $G_1(V, E)$ is a directed acyclic weighted graph in which each vertex $v_i \in V$ refers to an IP core (microprocessor, DSP and varieties of special function modules, etc.) and each directed arc $e_{i,j} \in E$ denotes a communication trace from a source vertex v_i to a destination vertex v_j . The weight of the arc $w_{i,j}$ represents the communication volume and $b_{i,j}$ is communication bandwidth requirement between these two vertices. For simplicity, this definition assumes implicitly that application tasks are already assigned to IP cores of the 3D NoC platform.

Definition 2 An topology architecture graph (TAG) $G_2(R, P)$ is a directed graph where each vertex $r_i \in R$ refers to a node of the 3D NoC and each arc $p_{i,j} \in P$ represents a physical link and $h_{i,j}$ represents Manhattan distance between r_i and r_j . $B_{i,j}$ is the largest communication bandwidth of $p_{i,j}$.

Using these two definitions above, the mapping problem can be formulated as follow.

Given an APCG G_1 and a TAG G_2 , find a map function $map()$ which maps each IP core $v_i \in V$ in G_1 to a node $r_i \in R$ in G_2 , so that the objective function is optimized and meets the following constraints:

$$\forall v_i \in V \Rightarrow map(v_i) \in R, \quad (1)$$

$$\forall v_i \neq v_j \Leftrightarrow map(v_i) \neq map(v_j), \quad (2)$$

$$size(G_1) \leq size(G_2), \quad (3)$$

$$\forall b_{i,j} \leq B_{i,j}. \quad (4)$$

In this paper, our goal is to minimize the power consumption and the corresponding objective functions described in the next section.

2.2 Mapping examples of 3D NoC

For each resource node can send data to any other resource node, IP core in G_1 can be mapped to any available resource node in G_2 . We assume the mapped relationship as $X = (x_1, x_2, \dots, x_n)$, in which n is the number of IP cores in G_1 . X is encoded as an arrangement of 1 to n , and a mapping scheme

can be obtained by decoding the encoding. Classic application characterization graph MWD [15] contains 12 nodes as shown in Fig. 1. IP cores in the figure are represented as 1 to 12. We explain the mapping process by taking MWD as an example and mapping IP cores in MWD to a 3D Mesh architecture of $2 \times 2 \times 3$.

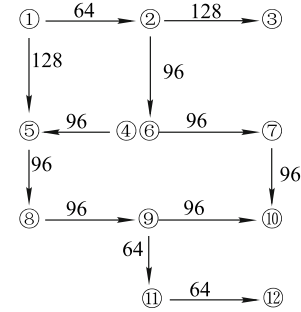


Fig. 1 Application characterization graph MWD [15]

Resource nodes on 3D NoC are represented as t_0 to t_{11} as shown in Fig. 2. t_0 to t_3 represent the nodes on the first layer, t_4 to t_7 represent the nodes on the second layer and t_8 to t_{11} represent the nodes on the third layer. $X=(3,2,4,1,5,7,6,9,12,8,10,11)$ represents mapping of IP core set (3,2,4,1,5,7,6,9,12,8,10,11) to a set of processing unit ($t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}$) and the mapping result is shown in Fig. 3.

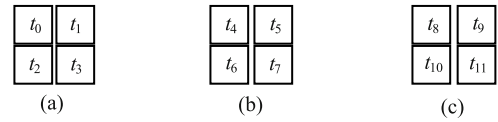


Fig. 2 Topologies of 3D Mesh. (a) Layer 1; (b) Layer 2; (c) Layer 3



Fig. 3 Mapping result

2.3 Power consumption model

Using the power consumption model given in Ref. [16], power consumption of the process of node t_i sending 1 flit to node t_j can be formulated as:

$$E_{bit}^{t_i, t_j} = \mu E_{Rbit} + \mu_H E_{LHbit} + \mu_V E_{LVbit}. \quad (5)$$

In Eq. (5), μ represents the number of routers between t_i to t_j . μ_H is the number of edges between t_i to t_j in horizontal direction and μ_V is the number of edges between t_i to t_j in vertical direction. E_{Rbit} represents the power consumption of a flit through a router. E_{LHbit} represents the power consumption of a flit through circuit in horizontal direction and E_{LVbit} represents the power consumption of a flit through circuit in

vertical direction. The computational formulae of E_{LHbit} and E_{LVbit} as given in Ref. [17] are as follows:

$$E_{LHbit} = d_H V_{dd}^2 C_{wireH} / 2, \quad (6)$$

$$E_{LVbit} = d_V V_{dd}^2 C_{wireV} / 2. \quad (7)$$

In Eq. (6), d_H represents the length of circuit and C_{wireH} represents the capacitance of circuit in horizontal direction. In Eq. (7), d_V represents the length of circuit and C_{wireV} represents the capacitance of circuit in vertical direction. V_{dd} represents supply voltage.

Mapping of 3D NoC is to find an association between IP cores and resource nodes. Find a mapping function under the conditions of a given APCG and TAG, so that the total power consumption is the lowest and the following conditions are met:

$$\min \left(\sum_{v_i, v_j \in V} (w_{i,j} \times E_{bit}^{map(v_i), map(v_j)}) \right), \quad (8)$$

$$\forall v_i \in V, map(v_i) \in R, \quad (9)$$

$$\forall v_i \neq v_j, map(v_i) \neq map(v_j). \quad (10)$$

For a determined topology, E_{Rbit} , E_{LHbit} and E_{LVbit} are determined. According to Ref. [17], the power consumption on the circuit is proportional to the length of the circuit. So the length of the circuit between two communicating nodes is a key influence affecting the power consumption of 3D NoC. Power consumption of 3D NoC in vertical direction is much less than that in horizontal direction when the same amount of data is transferred due to usage of TSV (Through Silicon Vias) technology.

3 Mapping of 3D NoC based on PSO and QPSO algorithm

Currently, most of the mapping algorithms of 3D NoC are heuristic mapping algorithms, and there are also some non-heuristic mapping algorithms. In heuristic mapping algorithms, mapping algorithms based on particle swarm optimization algorithm are very representative.

3.1 PSO algorithm and QPSO algorithm

3.1.1 PSO algorithm

Particle swarm optimization (PSO) algorithm was proposed by American social psychologist James Kennedy and electrical engineer Russell Eberhart in 1995 [18]. They modified the model of imitating flock [19] proposed by Hepper, so that the particles can fly to the solution space and land on the best solution, resulting in PSO algorithm. The algorithm is

an optimization algorithm based on swarm intelligence and obtains best solution by the interaction among particles [20]. The algorithm is simple and easy to realize without gradient information and has yielded good results for both continuous optimization and discrete optimization problems. The feature of natural real coding makes PSO suitable for handling real optimization problems.

3.1.2 QPSO algorithm

Quantum-behaved particle swarm optimization (QPSO) algorithm is proposed by Sun Jun from Jiangnan university of China [21]. There are some flaws of PSO algorithm. First, PSO is not a global convergence algorithm in theory [22]. Secondly, the evolutionary formula of speed and position of PSO leads to low randomness and intelligence. In addition, reliance of algorithm performance on speed leads to low robustness. In response to these shortcomings, Sun studied the basic characteristics of swarm intelligence and human learning model in-depth and found that human's intelligent behavior is very similar to the behavior of particles in quantum space. Quantum system has a strong uncertainty due to the superposition of states and the human mind is also uncertain. Thus it is logical to describe human thinking and intelligence with the quantum model. Sun set up a particle swarm model based on δ potential of quantum and proposed QPSO algorithm. QPSO is a global convergence algorithm and has many advantages such as global convergence, less control parameters, faster convergence and stronger ability of optimization.

3.2 Theory of mapping algorithms of 3D NoC based on PSO and QPSO

3.2.1 Theory of mapping algorithm of 3D NoC based on PSO

The fitness function of mapping algorithm based on PSO used is shown in Eq. (11):

$$Cost = MAXFit - (d_x \times w_x + d_y \times w_y + d_z). \quad (11)$$

In Eq. (11), $MAXFit$ is the biggest fitness value set for in advance. d_x , d_y and d_z are respectively represent the distance between communication nodes in the x -axis, y -axis and z -axis. w_x and w_y respectively represent traffic between nodes in the x -axis and y -axis. Power consumption of 3D NoC in vertical direction is much less than that in horizontal direction when the same amount of data is transferred due to TSV technology involved [23,24]. Therefore, in the z -axis direction, it is no longer multiplied by the weight. From Eq. (11), we can know that the fitness value is inversely proportional

to the traffic of 3D NoC. The greater the traffic, the lower the fitness value. We can know from the power consumption model above that the greater the traffic, the higher the power consumption. Thus the power consumption brought by the algorithm can be judged by the size of the fitness value, i.e., the greater the fitness value, the lower the power consumption. The evolutionary formulae of speed and position of the algorithm are shown in Eqs. (12) and (13):

$$v_{i+1} = \omega v_i + c_1 \xi (\text{indibest}_i - x_i) + c_2 \eta (\text{best}_i - x_i), \quad (12)$$

$$x_{i+1} = x_i + v_i. \quad (13)$$

In Eq. (12), velocity v_i and position x_i are sequences with DIM dimension. DIM is the number of IP cores of APCG. The velocity is in the range $[-DIM/4, DIM/4]$. If the value is larger than $DIM/4$, then it can be revised to equal to the $DIM/4$. If the value is smaller than $-DIM/4$, then it can be revised to equal to the $-DIM/4$. The position sequence of particles is the sequence of IP cores of APCG on resource nodes of 3D NoC. The position is in the range $[1, DIM]$ and the values in one sequence are different from each other. If the value in sequence x_{i+1} is the same with one another, then replace one of them with a value in $[1, DIM]$ that was not used and ensure the value which is replaced is closer with the number which replaces it. The inertia weight ω is variable weight and the calculated formula is shown as Eq. (14). The initial value of ω_{\min} and ω_{\max} are 0.5 and 1. The parameter *count* represents the current number of iterations and N represents the maximum number of iterations. Learning factors c_1 and c_2 use synchronous time-varying way and the calculated formula is defined as Eq. (15). The initial value of c_{\min} and c_{\max} are 0.25 and 3. Random parameters d_1 and d_2 are numbers that program randomly generated between 0 and 1. The parameter *indibest* _{i} represents the best position of the particle itself, i.e., the sequence with best fitness value in the iteration process of the particle. *best* _{i} represents the optimum position of particle swarm, i.e., the sequence with best fitness value in the iteration process of the particle swarm.

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min}) \times \text{count}}{N}. \quad (14)$$

$$c_1 = c_2 = c_{\max} - \frac{(c_{\max} - c_{\min}) \times \text{count}}{N}. \quad (15)$$

3.2.2 Theory of mapping algorithm of 3D NoC based on QPSO

In this paper, QPSO algorithm is applied to the mapping problem of 3D NoC for the first time. QPSO algorithm is used to optimize the mapping position of IP cores of APCG on the resource nodes of 3D NoC.

The fitness function of mapping algorithm based on QPSO used is shown in Eq. (11). The greater the fitness value, the lower the power consumption produced in mapping. The evolutionary formula of position of the algorithm is shown in Eq. (16).

$$x_{i,j}(t+1) = p_{i,j}(t) - \alpha \times |c_j(t) - x_{i,j}(t)| \times \ln(1/\mu). \quad (16)$$

In Eq. (16), position $x_i(t)$ is sequence with DIM dimension. DIM is the number of IP cores of APCG.

The position sequence of particles is the sequence of IP cores of APCG on resource nodes of 3D NoC. The position is in the range $[1, DIM]$ and the values in one sequence are different from each other. If the value in sequence $x_i(t+1)$ is the same with one another, then replace one of them with a value in $[1, DIM]$ that was not used and ensure the value replaced is closer to the number replacing it. $p_{i,j}(t)$ represents the position of a random node and the calculated formula is shown as Eq. (17). φ is a number that randomly generated between 0 and 1. The parameter *indibest* _{i,j} (t) represents the best position of the particle itself, i.e., the sequence with best fitness value in the iteration process of the particle. *best* _{i,j} (t) represents the optimum position of particle swarm, i.e. the sequence with best fitness value in the iteration process of the particle swarm. α is an expansion shrinkage factor and the calculated formula is defined as Eq. (18). The initial value of α_{\max} and α_{\min} are 1 and 0.5. The parameter *count* represents the current number of iterations and N represents the maximum number of iterations. $c_j(t)$ is the average optimal position of the particle swarm and the calculated formula is shown as Eq. (19). M is the size of particle swarm. μ is a number that randomly generated between 0 and 1.

$$p_{i,j}(t) = \varphi \times \text{indibest}_{i,j}(t) + (1 - \varphi) \times \text{best}_j(t). \quad (17)$$

$$\alpha = \alpha_{\max} - \frac{(\alpha_{\max} - \alpha_{\min}) \times \text{count}}{N}. \quad (18)$$

$$c(t) = (c_1(t), \dots, c_j(t), \dots, c_{DIM}(t)) = \frac{1}{M} \sum_{i=1}^M \text{indibest}_i(t) = \left(\frac{1}{M} \sum_{i=1}^M \text{indibest}_{i,1}(t), \dots, \frac{1}{M} \sum_{i=1}^M \text{indibest}_{i,j}(t), \dots, \frac{1}{M} \sum_{i=1}^M \text{indibest}_{i,DIM}(t) \right). \quad (19)$$

3.3 Design and realization of mapping algorithms of 3D NoC based on PSO and QPSO

3.3.1 Realization of mapping algorithm of 3D NoC based on PSO

In this part, we show the realization of mapping algorithm based on PSO. The steps are as shown in Algorithm 1.

Algorithm 1 Mapping algorithm of 3D NoC based on PSO**Input:** Particle swarm randomly generated**Output:** Optimum position *best* of particle swarm

- Step 1** Parameters initialization. Define the number of particles as MAX_NP and the maximum number of iterations as MAX_GENERS . Randomly generate MAX_NP initial particles X_0 and velocity V_0 which are in $[-DIM/4, DIM/4]$.
- Step 2** Calculate the fitness value of each particle according to Eq. (11). If the particle adapts better than *indibest*, then the value is assigned to *indibest* and update the best position of the individual particle.
- Step 3** If the fitness of *indibest* is better than *best*, then update the value of *best*.
- Step 4** Update the velocity and position of particles according to Eqs. (12) and (13).
- Step 5** Correct the obtained position and velocity information of particles according to the determined range of value of position and velocity so that they would not exceed the available space.
- Step 6** Decide whether the current iteration number reaches the maximum number MAX_GENERS . If it is reached, then output *best*, else jump to step 2 and iteration number adds 1.
- Step 7** Output *best* and end the algorithm.

3.3.2 Realization of mapping algorithm of 3D NoC based on QPSO

In this part, we show the realization of mapping algorithm based on QPSO which is proposed in this paper. The steps are as shown in Algorithm 2.

4 Experiments and performance analysis

4.1 Simulation platform and parameter design

- Simulation platform

In this paper, Access Noxim 0.2 developed by Kai-Yuan Jheng in National Taiwan University is adopted as our simulation software [25]. The Access Noxim is a co-simulation platform for 3D NoC system that couples the network model, power model and thermal model. They integrate Noxim (a cycle-accurate SystemC NoC simulator) and HotSpot (HotSpot provides the architecture-level thermal model), and adopt the power model of Intel's 80-core processor.

- Choice of topology and routing algorithm

3D Mesh architecture is formed by extending 2D Mesh directly to 3D space [1] and has a regular structure. The position of network nodes of 3D Mesh is homogeneous and the wiring is simple. TSV technology is used in the vertical direction which can reduce the overall wiring length. In our experiment, 3D Mesh is used in our simulations. As for routing algorithm, we adopt XYZ routing algorithm. In XYZ algorithm, data packets are passed first in the x -axis, then in the

y -axis, and finally in the z -axis. The algorithm is easy to understand and realize, and it is the most commonly used routing algorithm.

Algorithm 2 Mapping algorithm of 3D NoC based on QPSO**Input:** Particle swarm randomly generated**Output:** Optimum position *best* of particle swarm

- Step 1** Parameters initialization. Define the number of particles as MAX_NP and the maximum number of iterations as MAX_GENERS . Randomly generate MAX_NP initial particles.
- Step 2** Calculate the average optimal position of the particle swarm according to Eq. (19).
- Step 3** Calculate the fitness value of each particle according to Eq. (11). If the particle adapts better than *indibest*, then the value is assigned to *indibest* and update the best position of the individual particle.
- Step 4** If the fitness of *indibest* is better than *best*, then update the value of *best*.
- Step 5** For every dimension of the particle, calculate a random position according to Eq. (17).
- Step 6** Update the position of particles according to Eq. (16).
- Step 7** Correct the obtained position and velocity information of particles according to the determined range of value of position and velocity so that they would not exceed the available space.
- Step 8** Decide whether the current iteration number reaches the maximum number MAX_GENERS . If it is reached, then output *best*, else jump to step 2 and iteration number adds 1.
- Step 9** Output *best* and end the algorithm.

- Parameter setting

1) Parameter setting of the algorithms Population size is set as 200 and the number of iterations is set to 100.

2) Parameter setting of the simulation software Data packets are injected in the way of Memory-less Poisson distribution. Packet injection rate is set as 0.02. The total power consumption of 5000 cycles is summarized. The size of data packet is between 2 flits and 10 flits. The size of cache of each channel of the router is 8 flits.

4.2 Experiment results and performance analysis

In order to compare our proposed mapping algorithm of 3D NoC based on QPSO with the original mapping algorithm of 3D NoC based on PSO, we conduct the experiments of these two algorithms on the same simulation platform. To reflect the universality of the algorithms, we use both classic APCG and random APCG to do the experiment. Classic APCG includes PIP (8-core), MWD (12-core), VOPD (16-core) and DVOPD (32-core). Random APCG is generated by using Task Builder TGFF [26], and different APCGs have different numbers of IP cores. For each APCG, simulation experiments are conducted separately using mapping algorithms based on QPSO and PSO. Then the communication mode files with the best solutions are generated. Finally, the

communication mode files are fed into Access Noxim to start the simulation. In this paper, convergence speed and power consumption of the two algorithms are compared.

4.2.1 Comparison on convergence speed of mapping algorithms based on PSO and QPSO

- Comparison on convergence speed of the two algorithms for classic APCG

For classic APCG, comparison on convergence speed in 100 iterations of mapping algorithms based on PSO and QPSO is shown in Fig. 4, where the abscissa represents the number of iterations, the ordinate represents fitness value, and the solid circle represents the convergence point of the algorithm. From Fig. 4, we can observe that for any classic APCG, mapping algorithm based on QPSO can get optimal solutions within a few iterations and the convergence speed is faster than the mapping algorithm based on PSO obviously. Also, except for the PIP, optimal solution of mapping algorithm based on QPSO fits better than the optimal solution of mapping algorithm based on PSO. Because the number of IP cores of PIP is less, PSO algorithm is not easy to fall into local convergence and can get the same optimal solution with QPSO algorithm. It also can be seen that for APCG with large number of IP cores, the advantages of mapping algorithm based on QPSO are more obvious, i.e., it can get better solutions within a short period of time.

- Comparison on convergence speed of the two algorithms for random APCG

In this paper, random APCGs with 22-core, 43-core, 62-core, 83-core and 102-core are generated using TGFF. For different random APCGs, comparison on convergence speed in 100 iterations of mapping algorithms based on PSO and QPSO is shown in Fig. 5. From Fig. 5, we can observe that for any random APCG, convergence speed of mapping algorithm based on QPSO is faster than that of the mapping algorithm based on PSO. The final solution of mapping algorithm based on QPSO is also better than that of mapping algorithm based on PSO, and advantages are more obvious with increasing of the number of IP cores in APCG.

4.2.2 Comparison on power consumption of mapping algorithms based on PSO and QPSO

In our simulation experiments, for each APCG, the two algorithms separately circulates for ten times and the best individual is used to generate communication mode files.

- Comparison on power consumption of the two algorithms for classic APCG

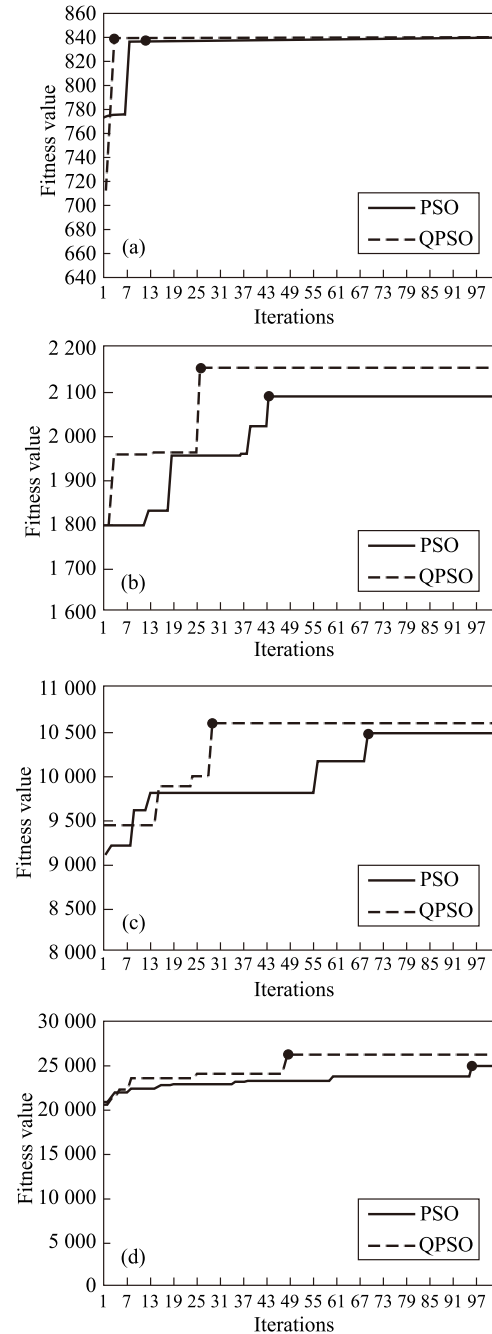


Fig. 4 Comparison on convergence speed for (a) PIP, (b) MWD, (c) VOPD, and (d) DVOPD

For different classic APCGs, mapping algorithm based on PSO circulates for ten times and the value of power consumption got from the simulation platform is shown in Table 1. Mapping algorithm based on QPSO circulates for ten times and the value of power consumption got from the simulation platform is shown in Table 2.

The experimental results are analyzed and the minimum power consumption, maximum power consumption and average power consumption of the two algorithms for the same

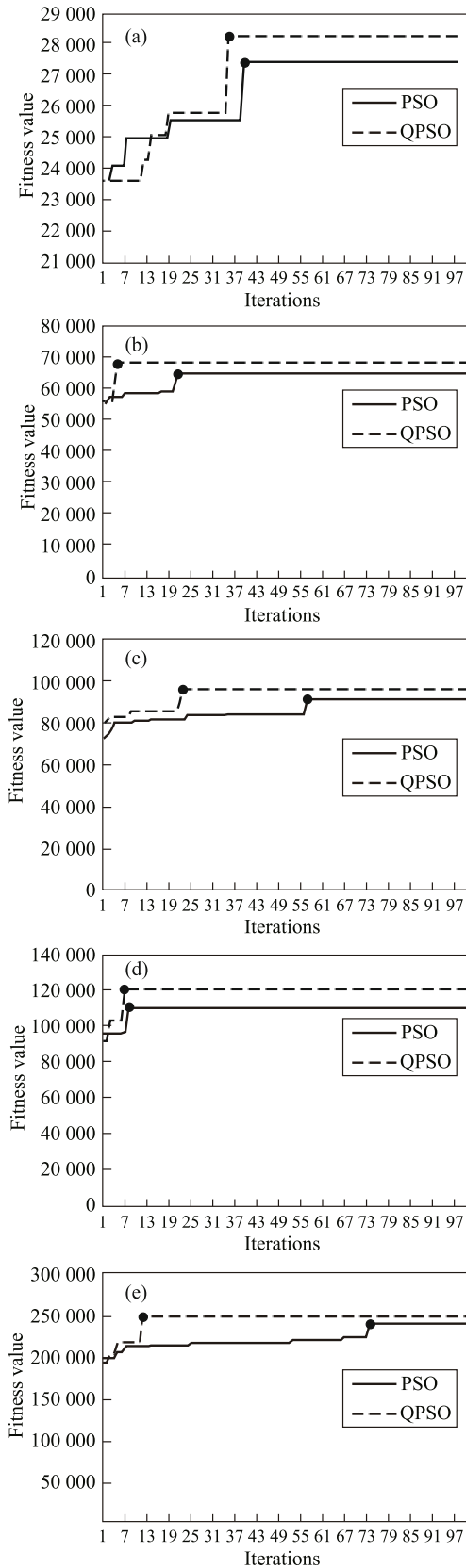


Fig. 5 Comparison on conversation speed for APCGs with different numbers of IP cores. (a) 22-core APCG; (b) 43-core APCG; (c) 62-core APCG; (d) 83-core APCG; (e) 102-core APCG

Table 1 Values of power consumption of mapping algorithm based on PSO of ten times

No.	Power consumption/J			
	PIP	MWD	VOPD	DVOPD
1	0.004 038 53	0.006 282 94	0.011 015 6	0.026 826 7
2	0.004 050 15	0.006 629 17	0.011 429 4	0.020 715 5
3	0.004 041 13	0.005 975 88	0.011 531 2	0.026 906
4	0.004 050 18	0.006 982 63	0.011 903 7	0.022 986
5	0.004 022 85	0.006 845 91	0.012 599 7	0.024 732 4
6	0.004 044 98	0.006 574 77	0.010 859 7	0.024 662 8
7	0.003 992 6	0.006 694 64	0.014 269 3	0.024 258 6
8	0.004 085 59	0.006 641 28	0.011 26	0.024 193 6
9	0.004 042 72	0.007 794 15	0.010 318 7	0.023 781 8
10	0.004 032 16	0.006 784 99	0.011 774 5	0.025 904 4

Table 2 Values of power consumption of mapping algorithm based on QPSO of ten times

No.	Power consumption/J			
	PIP	MWD	VOPD	DVOPD
1	0.003 985 42	0.005 646 99	0.010 432 4	0.023 757 5
2	0.004 025 27	0.006 812 48	0.011 131 3	0.020 235 7
3	0.004 066 38	0.007 266 12	0.010 870 6	0.022 427 2
4	0.004 018 08	0.007 430 89	0.011 029 4	0.025 009 4
5	0.004 078 97	0.007 414 03	0.009 392 3	0.025 147
6	0.004 024 03	0.006 000 24	0.011 96	0.024 062 8
7	0.004 039 13	0.006 369 77	0.010 544 6	0.023 499 7
8	0.004 075 01	0.006 940 53	0.013 052 1	0.025 118
9	0.003 989 51	0.005 253 46	0.010 520 5	0.025 969 7
10	0.004 001 63	0.005 461 01	0.011 358 1	0.019 688 8

APCG in ten simulation trials are separately compared. The comparison on the minimum power consumption of mapping algorithms based on PSO and QPSO is shown in Fig. 6. From Fig. 6 we can observe for all classic APCGs, the power consumption of mapping algorithm based on QPSO is lower than that of mapping algorithm based on PSO with maximum case as 13.75% (for MWD) and the minimum case as 0.81% (for PIP).

The comparison on the maximum power consumption of mapping algorithms based on PSO and QPSO is shown in Fig. 7. From Fig. 7 we can see that for all APCGs, the power consumption of mapping algorithm based on QPSO is lower than that of mapping algorithm based on PSO with the maximum case as 9.33% (for VOPD) and the minimum case as 0.16% (for PIP).

The comparison on the average power consumption of mapping algorithms based on PSO and QPSO is shown in Fig. 8. From Fig. 8 we can see that for all APCGs, the power consumption of mapping algorithm based on QPSO is lower than that of mapping algorithm based on PSO with the maximum case as 6.05% (for VOPD) and the minimum case as 0.24% (for PIP).

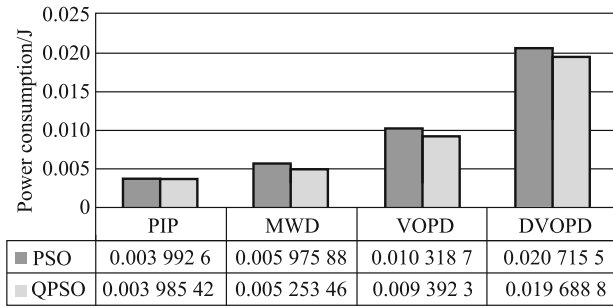


Fig. 6 Comparison on the minimum power consumption for classic APCG

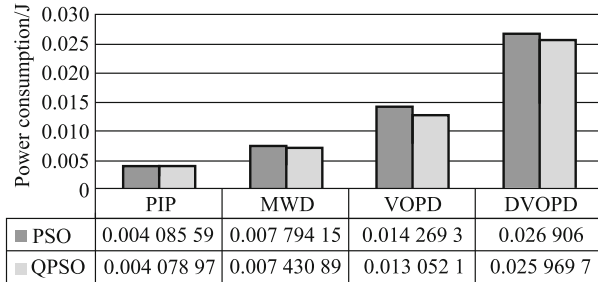


Fig. 7 Comparison on the maximum power consumption for classic APCG

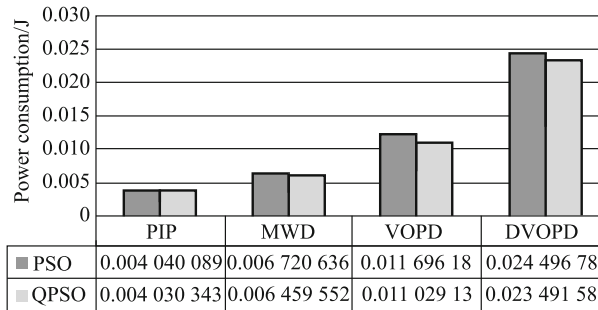


Fig. 8 Comparison on the average power consumption for classic APCG

In summary, it can be observed that power consumption is reduced in various degrees with mapping algorithm based on QPSO compared to the original mapping algorithm based on PSO. When the number of IP cores of APCG is small, PSO algorithm rarely falls in local optimum and the advantages of QPSO algorithm are not very obvious. When the number of IP cores of APCG is large, PSO algorithm tends to fall in local optimum and the ability of optimization of QPSO algorithm is stronger, especially it is easy to get very realistic minimum power consumption.

• Comparison on power consumption of the two algorithms for random APCG

In this paper, random APCGs with 22-core, 43-core, 62-core, 83-core and 102-core are generated using TGFF. For different random APCGs, mapping algorithm based on PSO circulates for ten times and the value of power consumption got from the simulation platform is shown in Table 3. Map-

ping algorithm based on QPSO circulates for ten times and the value of power consumption got from the simulation platform is shown in Table 4.

Table 3 Values of power consumption of mapping algorithm based on PSO of ten times

No.	Power consumption/J				
	22-core	43-core	62-core	83-core	102-core
1	0.004 038 53	0.006 282 94	0.011 015 6	0.026 826 7	0.041 210 3
2	0.004 050 15	0.006 629 17	0.011 429 4	0.020 715 5	0.047 018 2
3	0.011 979 1	0.021 020 5	0.029 891 1	0.042 805 4	0.041 436 8
4	0.010 384 75	0.021 437	0.030 129 7	0.040 874 8	0.041 305 5
5	0.011 214 3	0.020 199 9	0.027 932 7	0.040 186 8	0.043 047 4
6	0.012 187 2	0.021 445 6	0.030 725 8	0.043 086 5	0.042 426 8
7	0.012 232 1	0.023 533 2	0.026 857 7	0.040 621 3	0.038 294 7
8	0.012 233 6	0.021 096 1	0.031 588 8	0.046 018 2	0.049 163 9
9	0.012 235 2	0.021 141 2	0.031 098 1	0.040 807 7	0.044 028 8
10	0.010 986 8	0.019 460 8	0.030 851 3	0.041 758 2	0.043 105 9

Table 4 Values of power consumption of mapping algorithm based on QPSO of the times

No.	Power consumption/J				
	22-core	43-core	62-core	83-core	102-core
1	0.010 316 7	0.021 396 8	0.028 729 4	0.041 203 5	0.047 205 4
2	0.012 044 7	0.020 462 4	0.033 508 4	0.039 522 6	0.045 268 8
3	0.011 815 3	0.021 140 4	0.031 323 6	0.042 407 4	0.037 033 2
4	0.010 660 9	0.020 158 3	0.021 557 8	0.042 684 2	0.037 908 9
5	0.010 256 7	0.019 973 9	0.029 159	0.040 010 6	0.040 516 1
6	0.011 327 3	0.018 411 9	0.034 502 3	0.039 167 8	0.043 485 3
7	0.010 781 6	0.023 596 9	0.024 740 7	0.043 831 9	0.043 773 2
8	0.011 015 5	0.022 161 1	0.027 599 7	0.041 030 3	0.044 941 3
9	0.011 764 3	0.020 635 3	0.032 900 8	0.040 771 7	0.039 978 5
10	0.012 144 8	0.021 628	0.025 693 5	0.036 860 9	0.043 501

The experimental results are analyzed and the minimum power consumption, maximum power consumption and average power consumption of the two algorithms for the same APCG in ten simulation trials are separately compared. The comparison on the minimum power consumption of mapping algorithms based on PSO and QPSO is shown in Fig. 9. From Fig. 9 we can see that for all random APCG, the power consumption of mapping algorithm based on QPSO is lower than that of mapping algorithm based on PSO with the maximal decrease reaching 24.58% (for 62-core) and the minimum case as 1.25% (for 22-core).

The comparison on the maximum power consumption of mapping algorithms based on PSO and QPSO is shown in Fig. 10. From Fig. 10 we can see that for all random APCGs, the power consumption of mapping algorithm based on QPSO is lower than that of mapping algorithm based on PSO with the maximum case as 4.99% (for 83-core) and the minimum case as 1.81% (for 22-core).

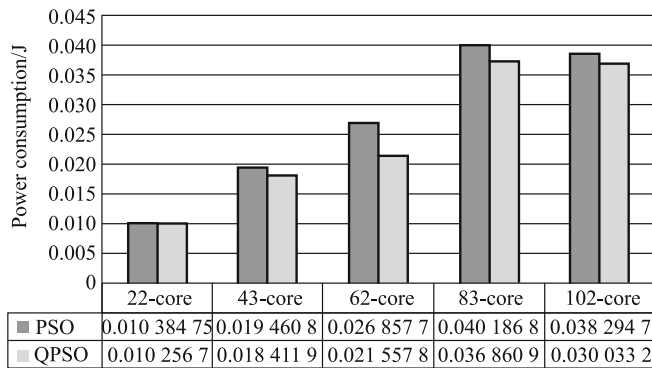


Fig. 9 Comparison on the minimum power consumption for random APCG

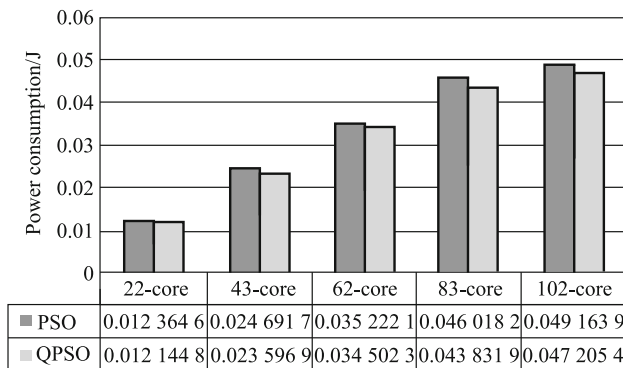


Fig. 10 Comparison on the maximum power consumption for random APCG

The comparison on the average power consumption of mapping algorithms based on PSO and QPSO is shown in Fig. 11. From Fig. 11 we can see that for all random APCGs, the power consumption of mapping algorithm based on QPSO is lower than that of mapping algorithm based on PSO with the maximum case as 14.43% (for 62-core) and the minimum case as 1.75% (for 102-core).

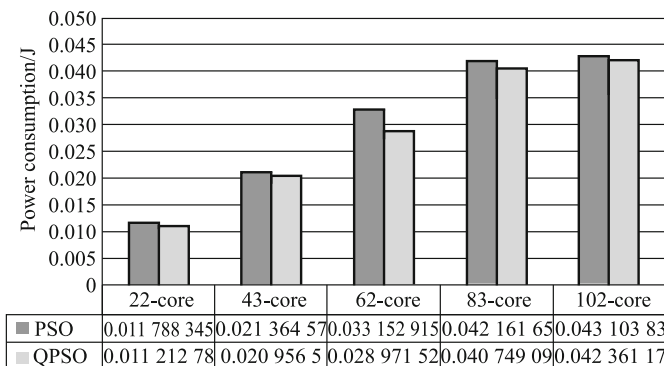


Fig. 11 Comparison on the average power consumption for random APCG

In summary, it can be observed that for random APCG, power consumption is reduced in varying degrees and the performance is more stable with mapping algorithm based on

QPSO compared to the original mapping algorithm based on PSO. This is because once the mapping algorithm based on PSO falls into a local optimum, it will lead to poor results. In addition, mapping algorithm based on QPSO is more suitable for APCG with about 60 IP cores.

5 Conclusions

In this paper, QPSO algorithm is used to solve the mapping problem of 3D NoC for the first time. Simulation results show that power consumption is reduced and the ability of optimization is improved, effectively reducing the power consumption of mapping of 3D NoC in the maximum case by 24.58% with mapping algorithm based on QPSO compared to the original mapping algorithm based on PSO. In the future, it is necessary to continue to improve QPSO algorithm and make it more suitable for mapping problem of 3D NoC. In addition, we will conduct research on mapping problem of 3D NoC for different topologies and routings, and make the proposed mapping problem based on QPSO be of more general applicability.

Acknowledgements This work was supported by the National Natural Science Foundation of China (NSFC) (Grant No. 61272006).

References

- Chen Y, Hu J, Ling X. Study on three-dimensional network on chip. *Telecommunications Science*, 2009, 25(4): 39–44
- Magarshack P, Paulin P G. System-on-chip beyond the nanometer wall. In: *Proceedings of the 40th Annual Design Automation Conference*. 2003, 419–424
- Dally W J, Towles B. Route packets, not wires: on-chip interconnection networks. In: *Proceedings of the 38th Design Automation Conference*. 2001, 684–689
- Benini L, De Micheli G. Networks on chips: a new SoC paradigm. *IEEE Computer*, 2002, 35(1): 70–78
- Kang A B. The ITRS design technology and system drivers roadmap: process and status. In: *Proceedings of the 50th Design Automation Conference*. 2013, 1–6
- Palesi M, Daneshtalab M. *Routing Algorithms in Network-on-Chip*. New York: Springer, 2014
- Xiang D, Liu G, Chakrabarty K, Fujiwara H. Thermal-aware test scheduling for NOC-based 3D integrated circuits. In: *Proceedings of the 21st IFIP/IEEE International Conference on Very Large Scale Integration*. 2013, 96–101
- Rahmani A M, Vaddina K R, Latif K, Liljeberg P, Plosila J, Tenhunen H. Design and management of high-performance, reliable and thermal-aware 3D network-on-chip. *IET Circuits, Devices & Systems*, 2012, 6(5): 308–321
- Hassanpour N, Hessabi S, Hamedani P K. Temperature control in

three-network on chips using task migration. *IET Computers & Digital Techniques*, 2013, 7(6): 274–281

10. Cheng Y, Zhang L, Han Y, Li X. Thermal-constrained task allocation for interconnect energy reduction in 3-D homogeneous MPSoCs. *IEEE Transactions on Very Large Scale Integration Systems*, 2013, 21(2): 239–249
11. Wang J, Li L, Yi W. A dynamic ant colony optimization algorithm for 3D NoC mapping. *Journal of Computer-Aided Design & Computer Graphics*, 2011, 23(9): 1614–1620
12. Wang J, Li L, Pan H, He S, Zhang R. Latency-aware mapping for 3D NoC using rank-based multi-objective genetic algorithm. In: *Proceedings of the 9th IEEE International Conference on ASIC*. 2011, 413–416
13. Sahni S, Gonzales T. P-complete approximation problems. *Journal of the ACM*, 1976, 23(3): 555–565
14. Yang W. Study on low-power mapping of network on chip. *Modern Computer*, 2015, 3(3): 10–13
15. Sahu P, Chattopadhyay S. A survey on application mapping strategies for network-on-chip design. *Journal of Systems Architecture*, 2013, 59(2013): 60–76
16. Yang W, Zhang Z, Liu Y. Improved particle swarm optimization algorithm based mapping algorithm for 3D-Mesh CMP. *Application Research of Computers*, 2013, 30(5): 1345–1348
17. Matsutani H, Koibuchi M, Amano H. Tightly-coupled multi-layer topologies for 3-D NoCs. In: *Proceedings of International Conference on Parallel Processing*. 2007
18. Kennedy J, Eberhart R C. Particle Swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. 1995, 1942–1948
19. Heppner F, Grenander U. A stochastic nonlinear model for coordinated bird rocks. *The Ubiquity of Chaos*, 1990
20. Wang D W, Wang J W, Wang H F, Zheng R, Guo Z. *Intelligent Optimization Methods*. Beijing: Higher Education Press, 2007
21. Sun J. Study on Quantum-Behaved Particle Swarm Optimization Algorithm. Jiangnan University, 2009
22. Van Den Bergh F. An analysis of particle swarm optimizers. *Particle Swarm Optimization*, 2002
23. Wang F. Analysis of key characteristics of through-silicon-via (TSV)-based three-dimensional integrated circuits (3D ICs). Dissertation for the Doctoral Degree. Xi'an: Xidian University, 2014
24. Kim J, Pak J S, Cho J, Song E, Cho J, Kim H, Song T, Lee J, Lee H, Park K, Yang S, Suh M, Byun K, Kim J. High-frequency scalable electrical model and analysis of a through silicon via (TSV). *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2011, 1(2): 181–195
25. Jheng K Y, Chao C H, Wang H Y, Wu A Y. Traffic-thermal mutual-coupling co-simulation platform for three-dimensional Network-on-

Chip. In: *Proceedings of International Symposium on VLSI Design Automation and Test*. 2010, 135–138

26. Dick R P, Rhodes D L, Wolf W. TGFF: task graphs for free. In: *Proceedings of the 6th International Workshop on Hardware/Software Code Sign*. 1998, 97–101



Cui Huang received the BS degree in software engineering from Tianjin Polytechnic University (TJPU), China in 2013. She is currently pursuing the MS degree in School of Computer Science and Software Engineering, TJPU. Her main research interest now is 3D Networks-on-Chip.



Dakun Zhang received PhD degree in electronic engineering from Northeastern University, China in 2004. Since 2005, she has been a professor with the Department of Computer, School of Computer Science and Software Engineering, Tianjin Polytechnic University, China. Prof. Zhang is a member of the editorial board of the core journal of “Computer Science and Exploration” and the director of Tianjin Institute of Graphic. She is also the reviewer of the core journal of “Journal of Chinese Computer Systems”. She has recently published more than 20 articles. Her research interests include 3D Networks-on-Chip, combinational algorithm design, etc.



Guozhi Song received the PhD degree in electronic engineering from Queen Mary, University of London, UK in 2009. Since 2012, he has been an associate professor with the Department of Networks, School of Computer Science and Software Engineering, Tianjin Polytechnic University, China. He is the author of more than 40 articles, and more than 10 inventions. His research interests include 3D Networks-on-Chip, heterogeneous wireless network integration and queueing theory. Dr. Song is a member of IEEE and ACM.