Bat Algorithm Based Low Power Mapping Methods for 3D Network-on-Chips

Jiazheng Li[1,2], Guozhi Song[1(✉)], Yue Ma[1], Cheng Wang[3], Baohui Zhu[1], Yan Chai[1], and Jieqi Rong[1]

[1] School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China
guozhi.song@gmail.com

[2] School of Information Sciences, University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA

[3] School of Information Science, Yunnan University, Kunming 650500, Yunnan, China

Abstract. Mapping a task graph as a distribution of Intellectual Property (IP) cores onto a Network-on-Chip (NoC) is a NP-hard problem that significantly affects the performance metrics of the whole system including power, delay, load balance and heat. Intelligence optimization algorithms are widely used to solve mapping problems. Bat Algorithm (BA), a novel metaheuristic algorithm mimicking hunting behaviors of bats, which has never been applied in NoCs, is used in low power mapping methods for 3D NoCs in this paper for the first time. The BA based mapping algorithm shows better performance than other mainstream mapping algorithms in terms of the optimization efficiency and power consumption. However, the concept of the basic BA has obvious disadvantages. To improve the basic BA, we propose a Group-Searching Bat Algorithm (GSBA) that can better utilize individual bats. This improved mapping algorithm performs much better than the traditional BA, especially when the scale of the application graph is large.

Keywords: Network-on-Chip · Mapping algorithm · Low power · Bat Algorithm · Parallel computing

1 Introduction

Along with the rapid development of VLSI technology, processing elements (PE) integrated on a single chip are becoming more. Network-on-Chip is a novel architecture of System-on-Chip proposed to adapt this tendency [1]. Key problems of NoCs are categorized into three parts, i.e. NoC infrastructure, communication paradigm, and application mapping [2]. From 2D to 3D NoCs, how to map applications onto NoCs is always a crucial issue attracting scholars all over the world. A good mapping algorithm can be helpful in reducing power consumption and execution time, balancing the load, and decreasing communication delay, thus improving the whole efficiency and performance of a system [3]. Mapping for NoCs is a NP-hard problem, so intelligence algorithms are excellent methods to solve it [4].

D. Du et al. (Eds.): NCTCS 2017, CCIS 768, pp. 277–295, 2017.
https://doi.org/10.1007/978-981-10-6893-5_21

Bats, one of the most fascinating species, have the amazing ability of using acoustic wave to locate their preys and avoid obstacles. This behavior is called echolocation. Inspired by echolocation of bats, Xin-She Yang proposed a novel metaheuristic algorithm-Bat Algorithm [5]. BA and BA-based methods has been widely used in numerical optimization, engineering optimization and other optimization areas.

However, the concept of BA has never been applied to solve NoC mapping problems. In this paper, we introduce the BA concept to 3D NoC mapping for the first time. The BA-based mapping algorithm shows better performance than other two mainstream heuristic mapping algorithms, i.e. Genetic Algorithm (GA) and Particle Swarm Algorithm (PSO). After careful analysis, we find that the basic BA has certain shortcomings. To increase the efficiency of BA, we propose an improved BA called GSBA. The experimental results show that GSBA is far superior to BA in terms of convergence and power consumption.

The rest of this paper is organized as followings. Section 2 introduces basic concepts and necessary formulation of 3D NoC mapping problem and process. The standard BA for numerical optimization is elaborated in Sect. 3. Based on the givens of previous sections, Sect. 4 applies the standard BA in 3D NoC mapping and compares it with GA and PSO. Summarizing and analyzing some flaws of basic BA, Sect. 5 proposes a parallel computing approach called GSBA to improve it, and does experiments and simulations to verify the efficiency of the improved algorithm. Finally, Sect. 6 gives the conclusion of this paper and expectation of future works.

2 Problem Formulation for 3D NoC Mapping

In this paper, each mapping process is based on the condition that the Application Characteristic Graph (ACG) to be mapped and the Topology Architecture Graph (TAG) of a NoC are given. We also assume that a node of the ACG represents an IP core operating certain tasks. Here an IP core could be any functional modules such as a microprocessor. The aim of mapping is to distribute the IP cores on the ACG onto the resource nodes on the TAG of the NoC in one-to-one correspondence properly, trying to find one arrangement that consumes the power as low as possible.

2.1 Definitions and Constraint Conditions

To clearly demonstrate the mapping problem, here we give three necessary definitions and according examples.

Definition 1. Given an ACG $G(V,E)$ where a node $v_i \in V$ represents an IP core identified by an integer, a directed edge $e_{i,j} \in E$ represents the communication path between cores v_i and v_j and the weight of this edge represents the communication traffic between the two cores. Figure 1 shows a classical ACG called Picture-In-Picture (PIP).

Definition 2. Given a TCG $T(R,P)$ where a node $r_i \in T$ represents a resource node (namely, a PE), a directed edge $p_{i,j} \in P$ represents the communication path between nodes r_i and r_j and $h_{i,j}$ represents the Manhattan distance between the two nodes.

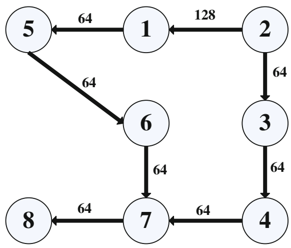


Fig. 1. Classical ACG PIP

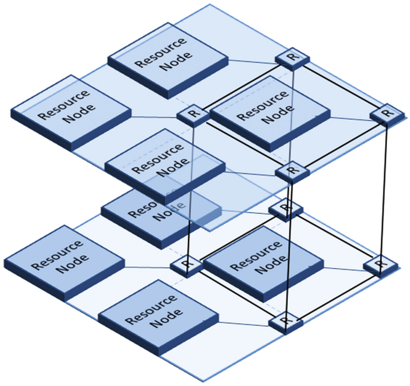


Fig. 2. A 2 × 2 × 2 mesh structure NoC

Figure 2 gives an example of a 2 × 2 × 2 Mesh structure NoC and Fig. 3 shows its according TAG.

Definition 3. Knowing an ACG with m cores and a TCG with n nodes (n ≥ m), one mapping assignment is represented by $(x_1, x_2, x_3, \ldots x_m)$ where x_i means putting the ith core of the ACG on the x_i th node on the TCG. Here we give an example of this real-number coding method: the mapping result of PIP ACG onto the 2 × 2× 2 NoC TAG shown in Fig. 3 is represented by (4, 1, 2, 3, 7, 6, 8, 5) (Fig. 4).

Based on the definitions above, the mapping process can be described as: given G and T, we try to find the mapping function $map()$ that would optimize the objective function. The mapping process must meet some specific constraint conditions as followings meanwhile.

$$\forall v_i \in V \Rightarrow map(v_i) \in R \tag{1}$$

$$\forall v_i \neq v_j \Leftrightarrow map(v_i) \neq map(v_j) \tag{2}$$

$$size(G) \leq size(T) \tag{3}$$

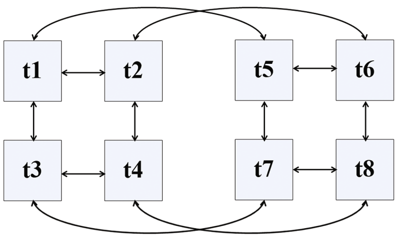


Fig. 3. The TAG of the NoC in Fig. 2

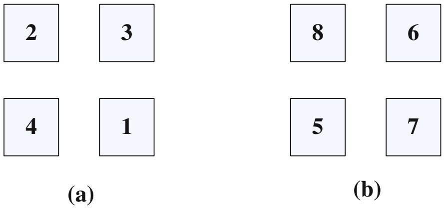


Fig. 4. One mapping result of PIP ACG onto the TAG in Fig. 3

2.2 Power Model for Mapping

The aim of optimized mapping methods in this paper is trying to reduce the total power consumption as much as possible. So power models for 3D NoCs are closely connected with the design of the objective function (i.e. the fitness function in the intelligence optimization process).

We use the power model given in [6]. The average energy consumed of transmitting one bit data form nodes r_i to r_j is calculated by

$$E_{bit}^{i,j} = \mu E_{Rbit} + \mu_H E_{LHbit} + \mu_V E_{LVbit} \tag{4}$$

where μ represents the number of routers from r_i to r_j, μ_H and μ_V represent numbers of horizontal and vertical links in the communication path respectively, and E_{LHbit} and E_{LVbit} represent consumed power of transmitting one bit data through one horizontal and one vertical link respectively.

The total power consumption thus would be represented as

$$E_{total} = \sum_{i=1}^{m} \sum_{j=1}^{m} (E_{bit}^{i,j} \times weight_{i,j}) \tag{5}$$

where $weight_{i,j}$ represents the communication traffic from r_i to r_j. This paper aims to minimize E_{total}.

3 Basic Concepts of Standard Bat Algorithm

Bats, the only mammal species with wings in the world, have amazing capability of echolocation. The frequencies of acoustic waves emitted by bats vary from different kinds of bats. Typical frequencies range from 25 to 150 kHz. Each ultrasonic burst typically lasts 5 to 20 ms. And a bat typically emits 10 to 20 bursts per second. When hunting and getting closer to their preys, the pulse emission rate can be sped up to about 200 pulses per second. The loudness of the pulses would normally decrease when flying to the preys. Modeling the echolocation behaviors of bat, BA was proposed [5]. Applied in various test problems, BA has shown its excellent efficiency in global engineering optimization [7]. BA performs nicely in terms of both local search and global search.

3.1 Global Search

In BA, we regard an individual bat as one solution of the optimization problem. The principle of BA global search is that other bats move towards the current best individual at some certain paces in one iteration. The bats change their positions according to the following rules.

$$V_i^{t+1} = V_i^t + \left(X_i^t - X_*\right) f_i \tag{6}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{7}$$

where V_i^t is the velocity of the ith individual bat at time t and X_i^t is its according position, X_* is the current global best position, and f_i is the frequency of this bat that controls how long one pace is calculated by

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{8}$$

where $[f_{min}, f_{max}]$ is the range of frequencies corresponding to the wavelength range $[\lambda_{min}, \lambda_{max}]$, and $\beta \in [0, 1]$ is a random number. The selection of f_i is related to the problem to be solved.

3.2 Local Search

The concept of BA local search is that one individual bat moves a small step around its neighborhood to look for a better solution. The movement of this individual bat can be mathematized as

$$X_{new} = X_{old} + \varepsilon \overline{A^t} \tag{9}$$

where $\varepsilon \in [-1, 1]$ is a random number and $\overline{A^t}$ is the average loudness of all bats at time t.

As in nature, the loudness A_i and the pulse emission rate R_i need to be updated during the optimization. As the iterations proceed, A_i decreases while R_i increases as

$$A_i^{t+1} = \alpha A_i^t \tag{10}$$

$$R_i^{t+1} = R_i^0[1 - \exp(-\gamma t)] \tag{11}$$

where α and γ are constants, and R_i^0 is the maximum pulse emission rate. Here α actually resembles the cooling factor in simulated annealing algorithm (SA). For any $0 < \alpha < 1$ and $\gamma > 0$, there is

$$ast \to \infty, A_i^t \to 0, R_i^t \to R_i^0 \tag{12}$$

3.3 Specific Steps

The specific steps of standard BA can be elaborated as followings.

Step (1) Confirm the objective function $f(X)$, the population scale and the number of iteration times.
Step (2) Randomly initialize the population and velocities V_i^0 of individuals.
Step (3) Use formula (8) to give each individual bat a constant frequency f_i.
Step (4) Initialize A_i^0 and R_i^0 for each individual bat.
Step (5) Find the current global best position X_* according to the fitness function.
Step (6) Update velocities and positions of individuals by formulas (6) and (7).
Step (7) For each individual bat, generate a random number $rand1_i$; if $rand1_i > R_i^t$, use formula (9) to accomplish the local search.
Step (8) Calculate the new fitness value of each individual.
Step (9) For each individual bat, generate a random number $rand2_i$; if $rand2_i < A_i^t$ and $f(X_{new}) < f(X_{old})$, accept the new solution.
Step (10) Increase R_i and decrease A_i using (10) and (11).
Step (11) Find the new global best individual.
Step (12) If the number of iteration times is met, output the best solution X_*; otherwise, jump to Step (6) and proceed the iteration.

4 Standard BA Based Mapping Method for 3D NoC

BA has been successfully applied in many domains, especially NP-hard problems, and shown its great efficiency. Wang et al. used BA in path planning for uninhabited combat air vehicle (UCAV) [8]. Bahman and Rasoul proposed an improved BA to develop strategies for Battery Energy Storage (BES) management [9]. Osaba et al. presented the first Discrete Bat Algorithm to solve Traveling Salesman Problem (TSP) and Asymmetric Traveling Salesman Problem (ATSP) [10]. Tangherloni et al. adopted BA in biochemical kinetic values parameter estimation [11].

However, such an efficient algorithm has never been used in the research on NoC design. Targeting at reducing power consumption of mapping applications onto 3D NoCs, we firstly try BA in the field of NoC.

However, we could not directly apply the standard BA, which is for continuous optimization problems, in 3D NoC mapping, a NP-hard combinatorial optimization problem with very strict constraint conditions. What we could do is to introduce the principle and the basic thought of BA to 3D NoC mapping, and properly modify them due to the particularity of our target problem.

4.1 Basic Idea

Heuristic algorithms have been the main methods in 3D NoC mapping [4]. The basic principle of all heuristic algorithms is generating a population group in which the individuals change following some special rules with various algorithms. We have mentioned in Sect. 2 that one mapping assignment is represented by $(x_1, x_2, x_3, \ldots x_m)$ where m is the number of IP cores to be mapped, and constrained that IP cores of an ACG and PEs on a TAG are in one-to-one correspondence. So one population of individuals used in our optimization is a set of mapping results.

In the iteration process, we use the searching principles of BA. To achieve global search, other non-best individuals are moving towards the current global best at a settled pace in each iteration. The "pace" here is different from one pace in numerical optimization. We define one pace movement of a non-best individual bat is changing one or two IP cores' positions, making the layout of this individual more alike with that of the current global best individual X_*. For a non-best individual X_i, this procedure is shown in Fig. 5 where we define $X_{i(k)}$ as the position of the kth IP core in the solution X_i.

And we also have the local search method. The procedure could be described as: randomly select an IP core and put it on another position of the NoC; if there is already an IP core on this position, swap the positions of the two IP cores. We call this shaking a mapping assignment.

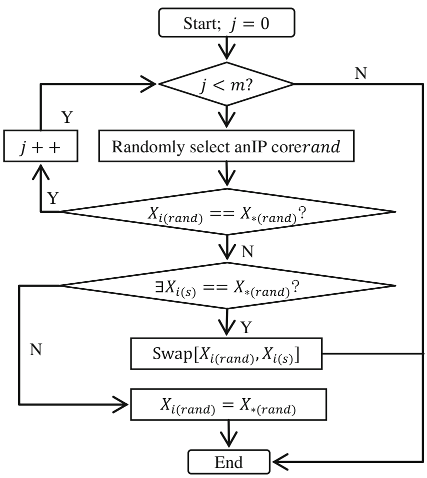


Fig. 5. Procedure of one pace movement of X_i

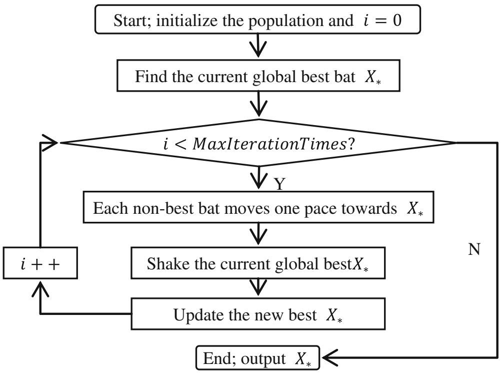


Fig. 6. BA based 3D NoC mapping algorithm

Combining the global search method and the local search method, we have the BA based optimization mapping algorithm for 3D NoC, as shown in Fig. 6.

4.2 Fitness Function Design

Actually, in the optimization process, only when the fitness value of one individual bat gets better, we accept the change. Fitness function is the criterion to evaluate a solution.

We extract our fitness function from the power model mention in formula (4) and (5). And we use XYZ routing algorithm [12] in our simulations. So the fitness function used in our optimization could designed as

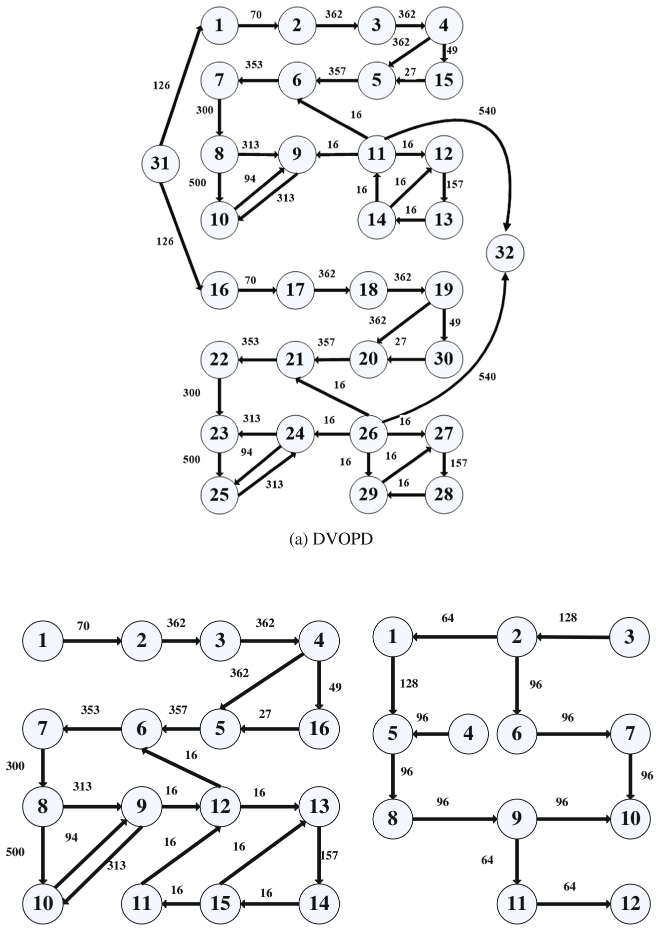
$$Fitness = \sum_{i=1}^{m}\sum_{j=1}^{m}\left[\left(X_{i,j} + Y_{i,j}\right)weight_{i,j} + Z_{i,j}\right] \tag{13}$$

where $X_{i,j}$, $Y_{i,j}$ and $Z_{i,j}$ are the numbers of links from core i to core j on x-axis, y-axis and z-axis respectively. Because of Through Silicon Vias (TSV) technology [13], the energy consumed vertically would be far less than the energy consumed horizontally. So $Z_{i,j}$ does not multiply by $weight_{i,j}$. From formula (13) we could see that the smaller the fitness value is, the better the solution is.

4.3 Comparison with GA and PSO on Convergence

After designing the algorithm and the fitness function. To verify the efficiency and superiority of BA based mapping algorithm, we conduct experiments comparing BA with GA [14] and PSO [15] under the same conditions. In this section, we use three classical ACGs: Multi-Window Displayer (MWD), Video Object Plane Decoder (VOPD) and Double Video Object Plane Decoder (DVOPD), as shown in Fig. 7.

To compare the three algorithms in terms of convergence, we run each algorithm for 10 times for each ACG, and we draw the average curve of 10 times experiments for clear description and avoiding deviation.



(a) DVOPD

(b) VOPD

(c) MWD

Fig. 7. Three classical ACGs

MWD has 12 cores and 14 edges. We map it onto a 2 × 2 × 3 mesh structure 3D NoC. And here the population of 3 algorithms is 50 and the number of max iteration times is 100. The average curves of 3 algorithms for mapping MWD are shown in Fig. 8.

VOPD has 16 cores and 21 edges. We map it onto a 3 × 3 × 3 mesh structure 3D NoC. And here the population of 3 algorithms is 100 and the number of maximum iteration times is 200. The average curves of 3 algorithms for mapping VOPD are shown in Fig. 9.

DVOPD has 32 cores and 44 edges. We map it onto a 4 × 4 × 3 mesh structure 3D NoC. And here the population of 3 algorithms is 200 and the number of max iteration times is 300. The average curves of 3 algorithms for mapping VOPD are shown in Fig. 10.

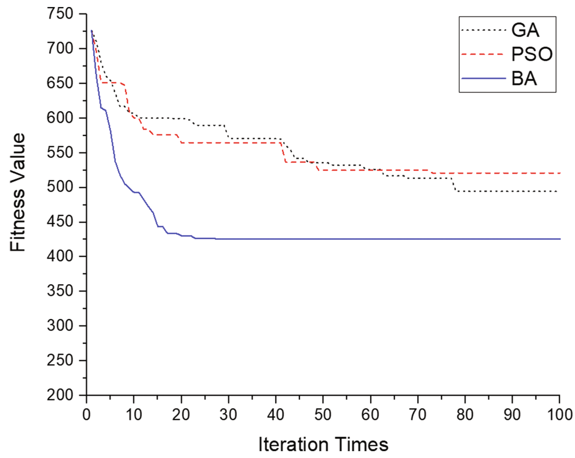


Fig. 8. Curves of 3 algorithms for mapping MWD

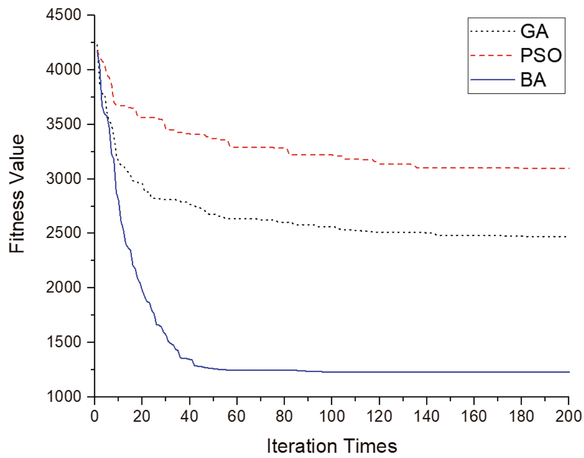


Fig. 9. Curves of 3 algorithms for mapping VOPD

From the mapping curves of 3 algorithms for 3 classical ACGs, we could see that for all the ACGs, BA could find better solutions with smaller fitness values. And for ACGs with larger scales, BA performs much better than other two algorithms.

4.4 Comparison with GA and PSO on Power Consumption

After getting the mapping results, to compare the power consumption situations, we generate communication files of the mapping results, and do simulations according the communication files on Access Noxim 0.2 platform [16]. Main simulation parameters are given in Table 1.

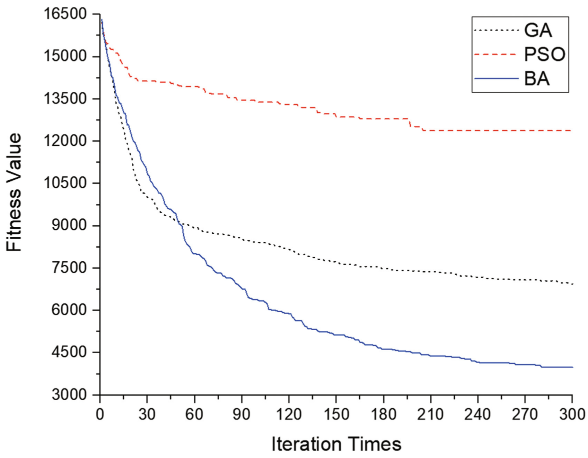


Fig. 10. Curves of 3 Algorithms for Mapping DVOPD

Table 1. Simulation parameters

Simulation parameter	Default value
Packet injection rate	0.02
Packet injection mode	Memory-less poisson distribution
Sizes of packets	2 to 10 flits
Channel buffer	8 flits

Setting the parameters, we do simulations using the final mapping results we obtained in the last part. For each ACG, the power consumption for every 10 mapping assignments of 3 algorithms are simulated and drawn in boxplots in the following Figs. 11, 12 and 13. Each calculated power consumed is for the whole system running an ACG.

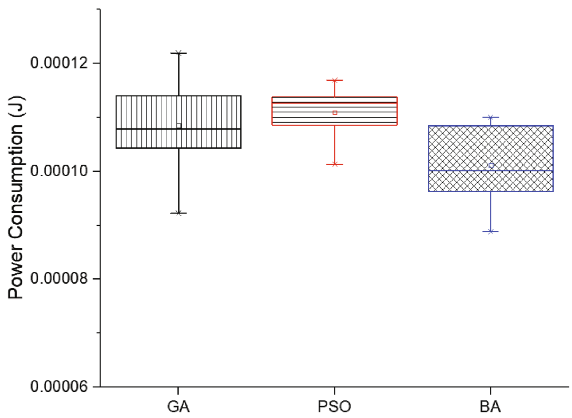


Fig. 11. Power consumption comparison for mapping MWD

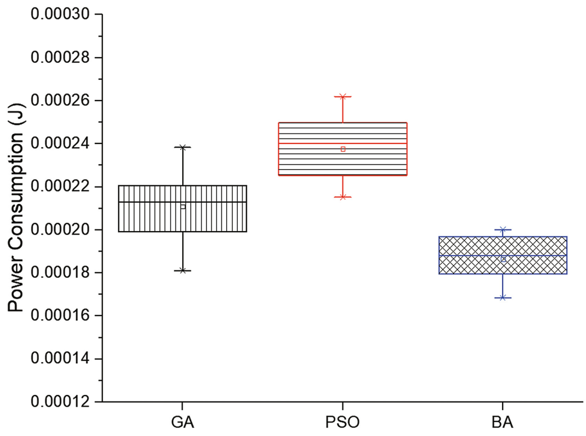


Fig. 12. Power consumption comparison for mapping VOPD

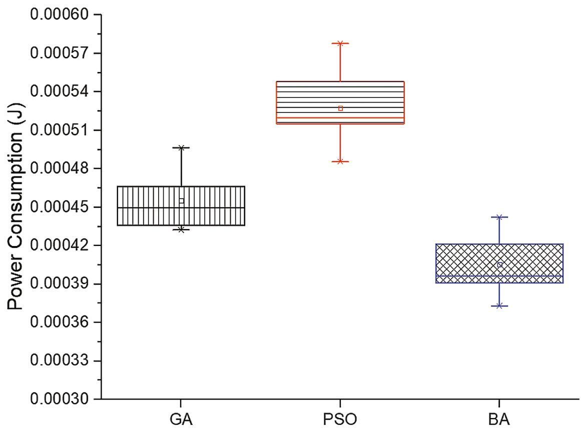


Fig. 13. Power consumption comparison for mapping DVOPD

We can observe that BA shows lower power consumption than GA and PSO. And the larger the scale of the ACG is, the more obviously BA excels other two algorithms. For MWD, the average power consumption of 10 times simulations of BA is 6.82% lower than that of GA and 8.91% lower than that of PSO. For VOPD, this figure of BA is 11.37% and 21.43% lower than those of GA and PSO respectively. For DVOPD, this figure of BA is 10.77% and 29.81% lower than those of GA and PSO respectively.

5 GSBA Based Mapping Method for 3D NoC

We successfully used standard BA in 3D NoC mapping in the last section. It has shown its great efficiency and searching ability. However, standard BA has certain defect like lack of diversity and instability of optimization accuracy. To improve the traditional BA, many approaches have been proposed. Fister et al. hybridized BA using Differential Evolution (DE) strategies [17]. Zhao and He presented a binary BA with chaos and Doppler effect [18]. Xue has done very deep research on BA and proposed several improvements [19]. And in this paper, aiming at increase the utilization of individuals and improve the diversity of populations, we propose a Group-Searching Bat Algorithm (GSBA).

5.1 GSBA Principles and Steps

Figure 14 is an example of showing one shortcoming of standard BA. For DVOPD, we use different population scale of BA for 10 times and draw the average curve.

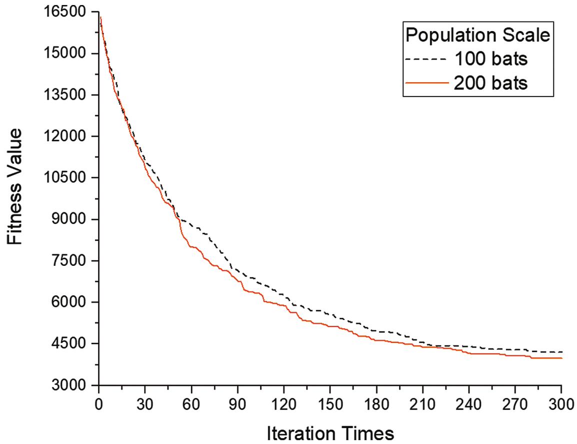


Fig. 14. 100-bat and 200-bat population scale for mapping DVOPD

It is obvious that the 200-bat population does not excel 100-bat population very much. This means that as the scale of population increase, standard BA cannot efficiently utilize each individual bat as we hope. The method that, all other bats move towards the current global best, lacks of diversity. To improve this, we introduce a group-searching method based on parallel computing into BA.

The basic principle of GSBA is that instead of only choosing the global best, we divide the population into several groups, and find the group best bats. In the iterations, each non-best individual in a group moves towards its group best. In this way, the bats in different groups fly to their own group best bat independently and simultaneously, thus the utilization of individual bats and diversity of the population are improved.

Except from this, to connect different groups and maintain the overall globality of the algorithm, after the group movements, we let each other group best individual move towards the global best. We also reinforce the local search to increase the searching accuracy. The specific steps of GSBA based mapping algorithm for 3D NoC are given in Fig. 15.

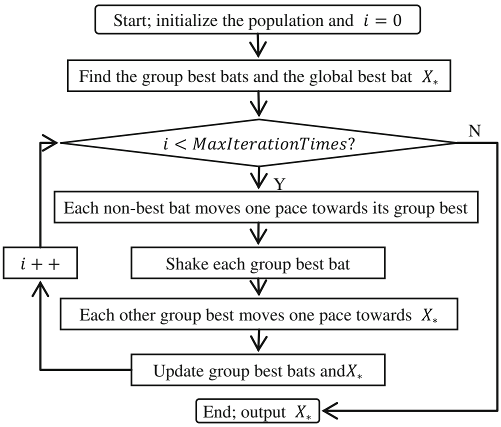


Fig. 15. GSBA based 3D NoC mapping algorithm

5.2 Experimental Results of Three Classical ACGs Comparing BA and GSBA

In this part, we do experiments and simulations to compare BA and GSBA in terms of convergence and power consumption by mapping three classical ACGs. Here the iteration times and population scales, and the NoCs structures are the same as Sect. 4.

To map MWD, the 50-bat population is divided into 5 groups. The convergence and power comparison between BA and GSBA are shown in Figs. 16 and 17.

To map VOPD, the 100-bat population is divided into 10 groups. The convergence and power comparison between BA and GSBA are shown in Figs. 18 and 19.

To map DVOPD, the 200-bat population is divided into 10 groups. The convergence and power comparison between BA and GSBA are shown in Figs. 20 and 21.

From these Figs, we could see that for MWD with the smallest scale, the optimization efficiency and power consumption of the two algorithms are almost the same. However, for VOPD and DVOPD with larger scales, the superiority of GSBA is obvious. In mapping VOPD, the average total power consumption of GSBA is 5.34% lower than that of standard BA. In mapping DVOPD, the average total power consumption of GSBA is 10.11% lower than its counterpart.

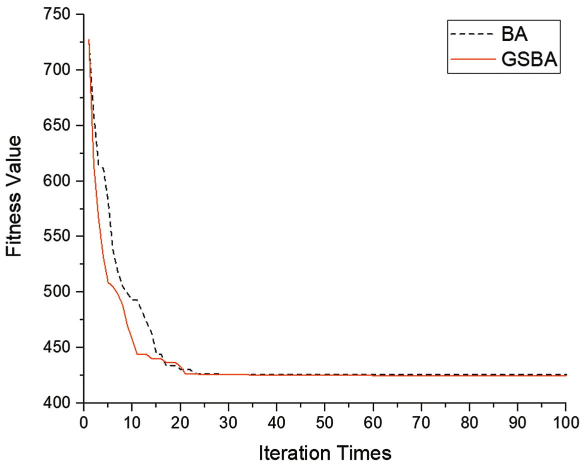


Fig. 16. Curves of BA and GSBA for mapping MWD

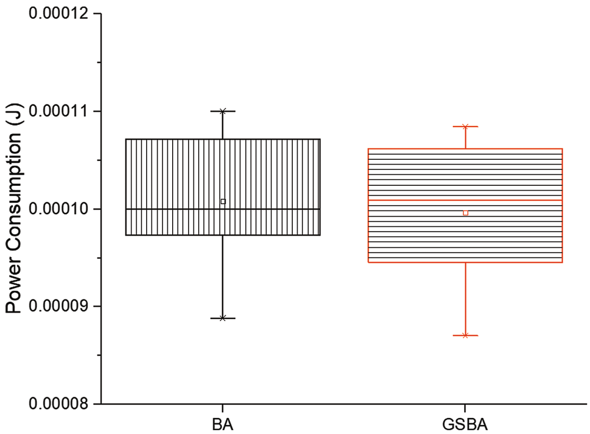


Fig. 17. Power consumption comparison for mapping MWD

5.3 Experimental Results of Five Random ACGs Comparing BA and GSBA

In this part, we do experiments and simulations to compare BA and GSBA in terms of convergence and power consumption by mapping five randomly generated ACGs. We use the random ACG generator TGFF [20] to make random ACGs in different scales. We generate 5 ACGs with 13 cores and 12 edges, 21 cores and 24 edges, 41 cores and 48 edges, 81 cores and 99 edges, and 101 cores and 129 edges respectively.

To map a random ACG with 13 cores and 12 edges, we use the $3 \times 3 \times 3$ Mesh structure heterogeneous NoC. The number of iteration times is 100 and the population scale is 50. We divide the 50-bat population into 5 groups.

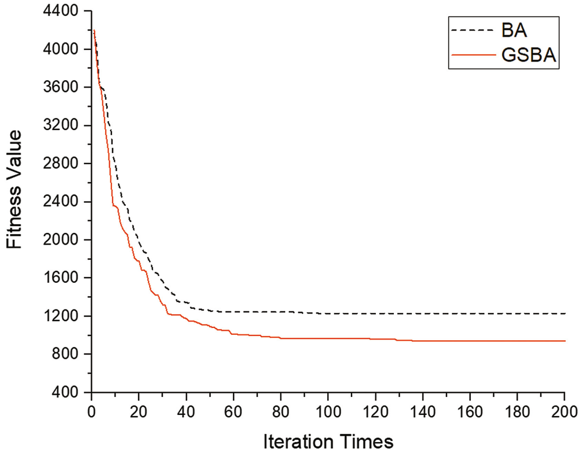


Fig. 18. Curves of BA and GSBA for mapping VOPD

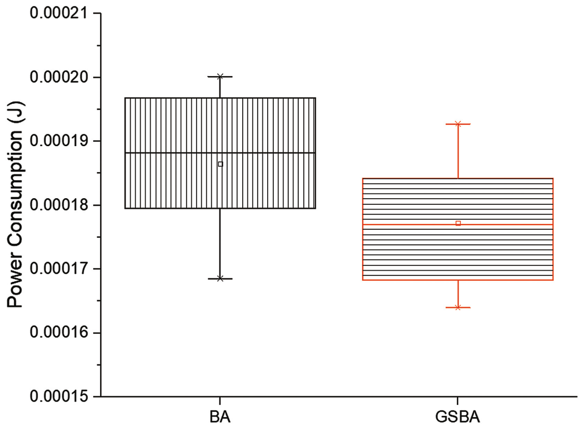


Fig. 19. Power consumption comparison for mapping VOPD

To map a random ACG with 21 cores and 24 edges, we use the 3 × 3 × 3 Mesh structure heterogeneous NoC. The number of iteration times is 200 and the population scale is 10. We divide the 100-bat population into 10 groups.

To map a random ACG with 41 cores and 48 edges, we use the 4 × 4 × 3 Mesh structure heterogeneous NoC. The number of iteration times is 200 and the population scale is 300. We divide the 200-bat population into 10 groups.

To map a random ACG with 81 cores and 99 edges, we use the 5 × 5 × 4 Mesh structure heterogeneous NoC. The number of iteration times is 300 and the population scale is 200. We divide the 200-bat population into 10 groups.

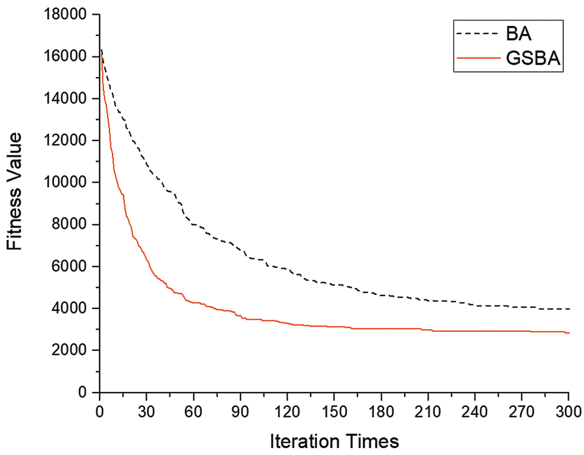


Fig. 20. Curves of BA and GSBA for mapping DVOPD

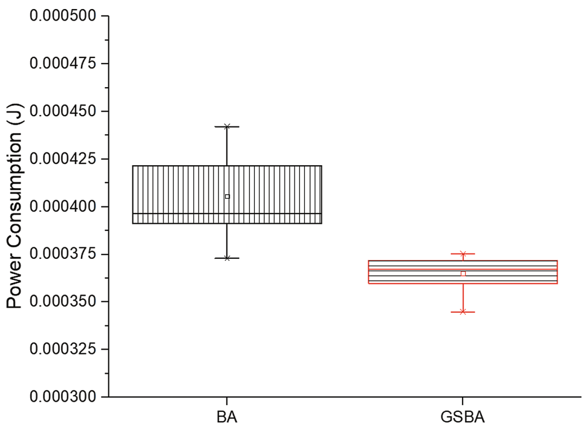


Fig. 21. Power consumption comparison for mapping DVOPD

To map a random ACG with 101 cores and 129 edges, we use the 5 × 5 × 5 Mesh structure heterogeneous NoC. The number of iteration times is 300 and the population scale is 200. We divide the 200-bat population into 10 groups.

For the random ACG in 13/12 scale, the average total power consumption of GSBA is 14.75% lower than that of standard BA. For the random ACG in 21/24 scale, that of GSBA is 10.32% lower than that of standard BA. For the random ACG in 41/48 scale, that of GSBA is 21.59% lower than that of standard BA. For the random ACG in 81/99 scale, that of GSBA is 30.16% lower than that of standard BA. For the random ACG in 101/129 scale, that of GSBA is 24.64% lower than that of its counterpart. Given the experimental data for five random ACGs in different scales, it is undoubted that GSBA has better efficiency and low power consumption than its counterpart, especially when the scale of the ACG is larger.

6 Conclusion and Future Work

In this paper, we successfully apply a novel heuristic algorithm called bat algorithm in low power mapping design methods for 3D NoC. This new algorithm performs better than other two mainstream mapping algorithms in terms of convergence and power consumption. However, we find that standard BA has certain weakness, thus proposing GSBA to increase the individual utilization and diversity of BA. Experimental results show that the improved BA with group-searching strategy is much superior to standard BA.

The limitation of this paper is that we optimize only one metric for 3D NoC. In our future work, we will focus on designing multi-objective optimization methods [21, 22], based on BA and its improvements, to optimize more than one performance metrics of power, heat, delay and load balance simultaneously.

And for the proposed GSBA, it is apparent that GSBA with different numbers of groups would perform very differently. In the future, we will explore the relationship between the efficiency and performance of GSBA and how to divide the population into groups by both experimental verification and theoretical support, trying to find the optimal group division method.

Acknowledgement. The authors were support by National Training Program of Innovation and Entrepreneurship for Undergraduates No. 201710058042 and 201710058009. We thank the anonymous reviewers for commenting on this paper.

References

1. Benini, L., De Micheli, G.: Networks on chips: a new SoC paradigm. Computer **35**(1), 70–78 (2002)
2. Ogras, U.Y., Hu, J., Marculescu, R.: Key research problems in NoC design: a holistic perspective. In: Proceedings of CODES+ISSS, Jersey City, NJ, pp. 69–74, September 2005
3. Hu, J., Marculescu, R.: Energy- and performance-aware mapping for regular NoC architectures. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **24**(4), 551–562 (2005)
4. Huang, C., Zhang, D.K., Song, G.Z.: Survey on mapping algorithm of three-dimensional network on chip. J. Chin. Comput. Syst. **37**(2), 193–201 (2016)
5. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: González, J.R., et al. (eds.) Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). SCI, vol. 284, pp. 65–74. Springer, Heidelberg (2010)
6. Wang, X.H., Liu, P., Yang, M., Palesi, M., Jiang, Y.T., Huang, M.C.: Energy efficient run-time incremental mapping for 3-D networks-on-chip. J. Comput. Sci. Technol. **28**(1), 54–71 (2013)
7. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
8. Wang, G.G., Guo, L.H., Duan, H., Liu, L., Wang, H.Q.: A bat algorithm with mutation for UCAV path planning. Sci. World J. (2012). Article ID 418946
9. Bahman, B.F., Rasoul, A.A.: Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm. Int. J. Electr. Power Energy Syst. **56**(3), 42–54 (2014)

10. Osaba, E., et al.: An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Eng. Appl. Artif. Intell. **48**(C), 59–71 (2016)
11. Tangherloni, A., Nobile, M.S., Cazzaniga, P.: GPU-powered bat algorithm for the parameter estimation of biochemical kinetic values. In: IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2016), Chiang Mai, Thailand, October 2016
12. Zhang, D.K., Huang, C., Song, G.Z.: Survey on three-dimensional network-on-chip. J. Softw. **27**(1), 155–187 (2016)
13. Black, B., Annavaram, M., Brekelbaum, N., DeVale, J., Jiang, L., Loh, G.H. et al.: Die stacking (3D) microarchitecture. In: IEEE/ACM International Symposium on Microarchitecture, pp. 469–479. IEEE Xplore (2006)
14. Addo-Quaye, C.: Thermal-aware mapping and placement for 3-D NoC designs. In: Proceedings of IEEE International SOC Conference, pp. 25–28. IEEE Xplore (2005)
15. Yang, W., Zhang, Z., Liu, Y.J.: Improved particle swarm optimization algorithm based mapping algorithm for 3D-Mesh CMP. Appl. Res. Comput. **30**(5), 1345–1348 (2013)
16. 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: International Symposium on VLSI Design Automation and Test, vol. 54, pp. 135–138. IEEE Explore (2010)
17. Fister, Jr., I., Fong, S., Brest, J., Fister, I.: A novel hybrid self-adaptive bat algorithm. Sci. World J. (2014). Article ID 709738
18. Zhao, D.S., He, Y.Z.: A novel binary bat algorithm with chaos and doppler effect in echoes for analog fault diagnosis. Analog Integr. Circ. Sig. Process **87**(3), 437–450 (2016)
19. Xue, F.: Research and application of heuristic intelligence optimization based on bat algorithm. Doctoral Dissertation, Beijing University of Technology (2016)
20. Dick, R.P., Rhodes, D.L., Wolf, W.: TGFF: task graphs for free. In: Proceedings of the Sixth IEEE International Workshop on Hardware/Software Codesign (CODES/CASHE 1998), pp. 97–101 (1998)
21. Tornero, R., Sterrantino, V., Palesi, M., Orduna, J.M.: A multi-objective strategy for concurrent mapping and routing in networks on chip. In: IEEE International Symposium on Parallel and Distributed Processing (IPDPS), Rome, Italy, May. 2009
22. Yang, X.S.: Bat algorithm for multi-objective optimisation. Int. J. Bio Inspir. Comput. **3**(5), 267–274 (2012)