WOAGA: A new metaheuristic mapping algorithm for large-scale mesh-based NoC

Xilu Wang, Yongjun Sun\textsuperscript{a),} Huaxi Gu, and Zujun Liu
State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an 710071, China
\textsuperscript{a)} yjsun@mail.xidian.edu.cn

Abstract: The mapping of IP cores to the topology is one of the most important steps for NoC (Network-on-Chip) design. Metaheuristic algorithms (MAs) are widely employed since the mapping is an NP-hard problem. Most mapping algorithms only consider small-scale NoC and ignore stability. In this letter, a stable metaheuristic algorithm called WOAGA, based on Whale Optimization Algorithm (WOA) and Genetic Algorithm (GA), is proposed for large-scale NoC mapping to achieve the low-energy consumption and stability. In the proposed algorithm, irregular crossover and mutation operations are integrated into the modified WOA. A perturbation is utilized to jump out of local optima effectively. Simulation results show that the proposed algorithm is more stable and achieve better solution with energy consumption reduced significantly.

Keywords: network-on-chip, mapping, metaheuristic algorithm

Classification: Integrated circuits

References

[1] L. Benini and G. De Micheli: “Networks on chips: A new SoC paradigm,” Computer 35 (2002) 70 (DOI: 10.1109/2.976921).
[2] M. Reshadi, et al.: “Elixir: A new bandwidth-constrained mapping for networks-on-chip,” IEICE Electron. Express 7 (2010) 73 (DOI: 10.1587/elex.7.73).
[3] G. Gharooni-fard, et al.: “Evaluating the performance of one-dimensional chaotic maps in the network-on-chip mapping problem,” IEICE Electron. Express 6 (2009) 811 (DOI: 10.1587/elex.6.811).
[4] T. Lei and S. Kumar: “A two-step genetic algorithm for mapping task graphs to network on chip architecture,” Proceedings of the Euromicro Symposium on Digital System Design 14 (2003) 180 (DOI: 10.1109/DSD.2003.1231923).
[5] Q. Q. Zhou, et al.: “Multi-objective mapping for network-on-chip based on bioinspired optimization algorithms,” Prognostics and System Health Management Conference (2014) 387 (DOI: 10.1109/PHM.2014.6988200).
[6] F. Moein-darbari, et al.: “CGMAP: A new approach to network-on-chip mapping problem,” IEICE Electron. Express 6 (2009) 27 (DOI: 10.1587/elex.6.27).
[7] M. Janidarmian, et al.: “Onyx: A new heuristic bandwidth-constrained...”
mapping of cores onto tile-based network on chip,” IEICE Electron. Express 6 (2009) 1 (DOI: 10.1587/elex.6.1).

[8] A. Mehran, et al.: “Spiral: A heuristic mapping algorithm for network on chip,” IEICE Electron. Express 4 (2007) 478 (DOI: 10.1587/elex.4.478).

[9] Y. Liu, et al.: “Energy and thermal aware mapping for mesh-based NoC architectures using multi-objective ant colony algorithm,” 2011 3rd International Conference on Computer Research and Development (ICCRD) (2011) 304 (DOI: 10.1109/ICCRD.2011.5764225).

[10] S. Saeidi, et al.: “Crinkle: A heuristic mapping algorithm for network on chip,” IEICE Electron. Express 6 (2009) 1737 (DOI: 10.1587/elex.6.1737).

[11] J. Hu and R. Marculescu: “Communication and task scheduling of application-specific networks-on-chip,” IEE Proceeding Computers and Digital Techniques 152 (2005) 643 (DOI: 10.1049/ip-cdt:20045092).

[12] T. Lei and S. Kumar: “Key research problems in NoC design: A holistic perspective,” Proc. CODES+ISSS05 (2005) 69C74, 19C21.

[13] S. Mirjalili and A. Lewis: “The whale optimization algorithm,” Advances in Engineering Software 95 (2016) 51 (DOI: 10.1016/j.advengsoft.2016.01.008).

[14] X. Y. Wang, et al.: “A novel heuristic algorithm for IP block mapping onto mesh-based networks-on-chip,” J. Supercomput. 72 (2016) 2035 (DOI: 10.1007/s11227-016-1719-6).

1 Introduction

With complexity increasing of the integrated circuits and the higher integration of Systems-on-Chip (SoC), the traditional bus is unable to meet the communication needs of multiple IP cores. Therefore, Network on Chip (NoC) has been created as a new communication framework [1]. One of the most critical steps in the NoC design process is mapping the IP cores on the topology of NoC to achieve better network performance [2].

Energy consumption on these integrated systems has been a critical issue. To achieve the best network performance, reduction of energy consumption has become a general purpose of NoC mapping. The mapping of NoC is a well-known NP-hard problem. Therefore, various researchers have presented metaheuristic or hybrid metaheuristic algorithms to solve the mapping optimization [3]. A two-step GA is proposed for mapping the applications onto a 2D mesh-based NoC [4], which only takes execution time into account. A Multi-Objective GA is used to optimize the mapping results [5]. Fahime Moein et al. [6] propose a new approach based on GA and the chaotic systems for small-scale NoC mapping. Majid Janidarmian et al. [7] present Onyx mapping algorithm on 2D mesh topology. Authors in [8] introduce a heuristic core mapping algorithm called Spiral to improve energy consumption. A multi-objective Ant Colony Optimization algorithm (ACO) is presented for optimizing energy consumption and hotspot temperature of NoC in [9].

In this letter, we propose a new metaheuristic mapping algorithm (WOAGA) for large-scale and mesh-based NoC architecture. The objective of the proposed algorithm is to reduce the energy consumption significantly for large-scale NoC and improve the stability.
2 Problem formulation

2.1 Mapping model

The mapping looks for a one-to-one corresponding relation between the IP cores and tiles in architecture. In this section, we refer to the following weighted directed graphs to formulate this problem [10].

Definition 1: An Application Characterization Graph (APCG), is a directed weighted graph $G(C, A)$, where each vertex $c_i \in C$ denotes an IP core, the directed arc $a_{ij} \in A$ denotes a communication trace from the IP core $c_i$ to the IP core $c_j$, and the weight of each edge $w_{ij}$ indicates the communication volume on edge $a_{ij}$.

Definition 2: An Architecture Characterization Graph (ACG), is a directed graph $G(T, L)$, where each vertex $t_i \in T$ represents one tile which includes a router $r_i$, a network interface and a position of IP cores, the directed arc $l_{ij} \in L$ represents the routing path from $t_i$ to $t_j$.

Given an Application Characterization Graph $G(C, A)$ and an Architecture Characterization Graph $G(T, L)$, the mapping optimization is defined as follows:

$$\min \left\{ \sum_{c_i, c_j \in C} w_{ij} \times (\text{energy}(\text{map}(c_i), \text{map}(c_j))) \right\},$$

(1)

$$\forall c_i \in C, \text{map}(c_i) \in T,$$

(2)

$$\forall c_i \neq c_j \in C, \text{map}(c_i) \neq \text{map}(c_j) \in T,$$

(3)

where $\text{map}(c_i)$ and $\text{map}(c_j)$ denote the corresponding tiles $t_i$ and $t_j$ in NoC of IP cores $c_i$ and $c_j$, $T$ represents a collection of tiles, $\text{energy}(\text{map}(c_i), \text{map}(c_j))$ denotes the energy consumption for sending one bit of date from $t_i$ to $t_j$. Next subsection provides more details of the computation of energy consumption.

2.2 Energy consumption

In general, the purpose of mapping optimization is to minimize the total energy consumption over the communication links. The energy consumption of NoC represents the energy consumed by the routers and the links [11]. The energy consumed by the routers is given by:

$$E_{Rbit} = E_{Sbit} + E_{Bbit} + E_{Wbit},$$

(4)

where $E_{Sbit}$, $E_{Bbit}$ and $E_{Wbit}$ represent the energy consumed by switch, buffering and interconnection wires, respectively. Since the energy consumed by buffering $E_{Bbit}$ and internal wires $E_{Wbit}$ is negligible, the energy consumption model can be reduced to the Bit Energy [12] as follows:

$$E_{bit}^{t_i \rightarrow t_j} = (n_{hops} + 1) \times E_{Sbit} + n_{hops} \times E_{Lbit},$$

(5)

where $E_{bit}^{t_i \rightarrow t_j}$ denotes the energy consumption for sending one bit of data from $t_i$ to $t_j$, $E_{Sbit}$ and $E_{Lbit}$ represent the energy consumed by the switches and the links, respectively. And $(n_{hops} + 1)$ is the number of routers that a data traverses from the source tile to the destination tile. Specifically, $n_{hops}$ is evaluated by the Manhattan distance between $t_i$ to $t_j$ in a mesh. Consequently, the total energy consumption by the network communication is given by:
where $T$ is the total number of tiles in NoC, $w_{ij}$ denotes the communication volume over the communication links.

With the number of IP cores increasing, the conventional algorithms are not capable of tackling the complicated mapping problems. To address this problem, a hybrid algorithm called WOAGA is introduced in our work. The WOAGA algorithm achieves lower energy consumption and excellent stability for large-scale NoC mapping.

### 3 WOAGA algorithm

The original WOA [13] performs more effectively than some other biological-inspired algorithms for various kinds of optimization problems. However, it is easy to get trapped into local optima and result in low precision. Although WOA has advantages in handling continuous optimization problems, it can not be applied to discrete problems directly, such as the mapping of NoC. The WOAGA algorithm overcomes the disadvantages of WOA, and could solve the mapping of NoC effectively. The WOAGA algorithm retains the exploration mechanism implemented in WOA, which can achieve a better trade-off between exploration and exploitation. At the same time, a perturbation is utilized to jump out of local optima. In the meantime, the two-point crossover operator and mutation operator in parallel improve the precision and satisfy the constraints of the NoC mapping.

In WOAGA, the initial population is generated completely at random and consists a set of chromosomes. Each chromosome $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ represents a mapping solution, and the length of $X_i$ is the number of IP cores. The concrete value of the $j$-th gene denotes the place of the $j$-th IP core in mesh. By calculating the fitness values (energy consumption), the best chromosomes $X$ is saved for next new population. The control parameter $\gamma_{control}$ is decreased linearly from 2 to 0, and the coefficient $A$ and $\gamma_{control}$ are calculated as follows:

$$\gamma_{control} = (2 - t \times 2 / \text{MaxIter}),$$

$$A = \gamma_{control} \times (2 \times \text{rand} - 1),$$

here, $t$ is the number of iterations, MaxIter denotes the maximum number of iterations, and rand is a random number in [0,1].

For each iteration, the probability $p$, the control parameter $\gamma_{control}$ and the coefficient $A$ are updated. When $p > 0.5$, a perturbation produced by mutation is triggered to avoid getting stuck in local optima. The mutation can bring sudden changes into a solution and exploring a promising unexplored part of the search-space. When $p < 0.5$, WOAGA can choose either a stochastic or guided movement depending on $|A|$. When $|A| \geq 1$, the exploration phase is enforced. The irregular crossover and mutation operators are executed in parallel, then the better mapping solution with less energy consumption is selected as offspring individual. When $|A| < 1$, the exploitation phase is guided by the current best solution. For executing the irregular crossover operator, two mapping solutions (chromosomes) $X_{\text{rand}1}$ and $X_{\text{rand}2}$ are chosen randomly. The two-point crossover recombines $X_{\text{rand}1}$ and $X_{\text{rand}2}$. 

$$E_{\text{total}} = \sum_{i=1}^{T} w_{ij} \times E_{\text{bit}}^{i,j},$$

(6)
by exchanging a part of IP cores randomly, then amends duplicate genes to make sure the new solution is valid. The irregular crossover and mutation operators are shown in Fig. 1. The pseudo code of WOAGA algorithm is presented as below.

**Algorithm 1** WOAGA algorithm: a new metaheuristic algorithm based on GA and WOA

**Input:** The weighted matrix $G(C, A)$ of APCG

**Output:** The optimal mapping results, the minimum energy consumption

1: **Initialization**()
2: Initialize the population $X_i (i = 1, 2, \ldots, n)$ by integer string encoding
3: Calculate the fitness (energy consumption) of each search agent
4: $X^*$ = the best solution
5: **Main loop**()
6: **While** ($t < \text{MaxIter}$)
7: **For** each search agent
8: Update $\gamma_\text{control}$, $A$ and $p$
9: **If1** ($p < 0.5$)
10: **If2** ($|A| < 1$)
11: Crossover($X_{\text{rand}}$, $X^*$) and mutation($X^*$), chose the better one
12: **Else if2** ($|A| > 1$)
13: Crossover($X_{\text{rand}1}$, $X_{\text{rand}2}$) and mutation($X_{\text{rand}1}$), chose the better one
14: **End if2**
15: **Else if1** ($p > 0.5$)
16: Trigger a perturbation produced by mutation
17: **End if1**
18: **End for**
19: Check and repair duplicate genes
20: Update $X^*$ if there is a better solution
21: $t = t + 1$
22: **End while**
23: Return $X^*$ and the fitness of $X^*$

**4 Results and discussion**

In this section, the performance of WOAGA is compared with three types of traditional mapping algorithms, Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Random Algorithm (Rand). In order to valuate the effectiveness and the stability of the four algorithms, VOPD, MPEG-4, MMS, MWD and PIP are used as benchmarks. For the larger size, we employ random synthetic applications benchmarks generated by TGFF with difference sizes from $7 \times 7$ to $12 \times 12$.

To compare fairly, each mapping algorithm is executed 20 times for each target application. The minimum energy consumption and the average results are recorded. Randomization is often used in MAs so that stability is an important aspect of MAs. $\text{GAP}$ is generally used as a metric of stability [14]. The energy consumption saving is considered as a key performance indicator. The gap between the best results and the average results is described as follows:
where $k$ is equal to 20, and $E(X_i)$ represents the energy consumption of the NoC with the mapping solution $X_i$, $E(X_{\text{best}})$ denotes the minimum energy consumption obtained from 20 times executions.

Table I shows the average and the best energy consumption of the four algorithms for each target application. The GAP and the saving energy consumption percentages of various algorithms are also offered in Table I. Table II presents the improvement of WOAGA algorithm over GA, ACO and RAND.

For the five practical applications, WOAGA algorithm achieves the minimum energy consumption compared with the other three competitors. The energy consumption of WOAGA is reduced by 40% and above as compared to Random Algorithm. The energy consumption savings of WOAGA are between 10% and 60% when compared with GA and ACO. Note that the GAPs of WOAGA algorithm close to 0. It is obvious that WOAGA outperforms the compared algorithms in terms of stability and the energy consumption reduction. Concerning
the large-scale synthetic applications, WOAGA algorithm obtains the best stability and significant energy consumption improvement. Compared with GA and RAND, WOAGA achieves 50% or more improvements in energy consumption. The improvements of WOAGA over ACO are in the range of 35%–55%. Fig. 2 shows the energy consumption when the scale of NOC increases. In comparison to the other three methods, the proposed algorithm is very competent in solving large-scale mapping problems.

5 Conclusion

In this letter, the WOAGA algorithm is proposed to solve the mapping problem of NoC, mainly focusing on the low-energy consumption and large-scale mapping. WOAGA algorithm has only one control parameter and is easy to implement. The irregular crossover and mutation operations are executed in parallel to improve the precision. Remarkably, a perturbation is introduced to avoid premature convergence. Results show that WOAGA algorithm has excellent stability and reduces energy consumption significantly compared with other algorithms. The energy saving is more remarkable when the scale of NOC increases.

| Table II. Comparisons of Saving and GAP |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | WOAGA GAP%      | GA GAP%         | ACO GAP%        | RAND GAP%       | RAND Saving%    |
| VOPD            | 0.39            | 35.9            | 26.29           | 44.8            | 2.85            |
| MEPG            | 0.53            | 25.1            | 18.93           | 57.2            | 3.23            |
| MMS             | 0.65            | 41.5            | 14.40           | 50.1            | 1.31            |
| MWD             | 0               | 34.2            | 14.27           | 35.2            | 2.50            |
| PIP             | 0               | 14.7            | 17.21           | 12.1            | 4.35            |
| 7 × 7           | 1.66            | 50.8            | 5.13            | 38.4            | 4.90            |
| 8 × 8           | 1.62            | 56.7            | 4.16            | 42.2            | 5.24            |
| 9 × 9           | 2.82            | 57.1            | 5.33            | 54.7            | 5.73            |
| 10 × 10         | 2.3             | 60.6            | 6.52            | 46.4            | 6.91            |
| 11 × 11         | 2.42            | 60.6            | 8.77            | 45.6            | 7.62            |
| 12 × 12         | 2.71            | 57.6            | 3.56            | 47.7            | 8.25            |

Fig. 2. The energy consumption of the four algorithms for different sizes
Acknowledgments

This work was supported by the National Science Foundation of China under Grants 61571340, and Grant 61634004, the Fundamental Research Funds for the Central Universities Grant No. JB180309 and No. JB170107, the key research and development plan of Shaanxi province No. 2017ZDCXL-GY-05-01, the General Armament Department and Ministry of Education United Fund under Grant No. 6141A0224-003, and the program of Introducing Talents of Discipline to Universities Grant No. B0803.