# GA-MMAS: an Energy- and Latency-aware Mapping Algorithm for 2D Network-on-Chip

Ning Wu, Yifeng Mu, and Fen Ge

***Abstract*—In this paper, a new mapping algorithm named GA-MMAS is proposed based on Genetic Algorithm (GA) and MAX-MIN Ant System Algorithm (MMAS), using a unified cost function with energy and link load variance, to optimize energy consumption and latency for NoC. Firstly the proposed algorithm obtain the elicitation information via priority mapping of IP core with larger communication volume, instead of using heuristics, to improve the optimal solution of MMAS. Then with the combination of MMAS and GA, the advantage of speed in GA makes compensation to the lack of pheromone in the early stage of MMAS, and in turn enhancing the accuracy of optimal solution, which leads to lower energy consumption and latency. The experiments performed on various random benchmarks and a complex video/audio application to conform the efficiency of the algorithm. Experimental results show that when only optimizing energy consumption, the algorithm saves about 36%~60%, 3%~25%, 10%~30% and 3%~30% of energy consumption compared to random mapping, GA, Ant Colony Algorithm (ACA) and MMAS respectively. When only optimizing latency, the algorithm decreases about 36%~60%, 3%~25%, 10%~30% and 3%~30% of link load variances. When optimizing both of energy consumption and latency, the cost reducing results are 36%~60%, 3%~25%, 10%~30% and 3%~30%.**

***Index Terms* — energy consumption, latency, hybrid algorithm, mapping, NoC**

## I. Introduction

NETWORK-on-Chip(NoC) IP core mapping at system level significantly impacts the communication energy consumption and latency of the system [1]. Optimization of IP mapping can lead to significant energy savings and latency reducing.

NoC IP mapping is essentially a quadratic assignment problem which belongs to NP-complete problem. It is difficult to find an optimal result in reasonable time. In [2, 3, 4], Genetic Algorithm (GA) is applied to optimize delay, area or energy consumption. However GA will make some useless iteration in later period because of the underutilization of the feedback information. Ant Colony Algorithm (ACA) in [5] and MAX-MIN Ant System Algorithm (MMAS) in [6] are applied to optimize the energy consumption for IP mapping. However ACA or MMAS will find the solution very slowly because of the lack of initial pheromone.

In this paper, an algorithm named GA-MMAS is proposed for the problem of IP mapping based on researching on GA and MMAS. GA-MMAS firstly improves MMAS. Then it combines MMAS with GA to make compensation to their disadvantages by their advantages. Some of our initial work has been presented in [7] to optimize communication energy consumption. However, more work needs to be done in multi-objective optimization. Therefore, we use a unified cost function with energy and link load variance to optimize energy consumption and latency. Several experiments are carried out to verify the efficiency of the algorithm.

The rest of the paper is organized as follows: Section II describes the problem of IP core mapping; Section III illustrates the improved MMAS; Section IV illustrates our algorithm; Section V presents the experimental results; and finally, Section VI concludes the paper and outlines some directions for future work.

## II. Problem Descriptions

The problem of energy- and latency-aware mapping for NoC defines as follow: given the IP communication graph and NoC topology architecture, the designer decides which tile each IP should be placed to minimize the communication energy and latency of system. In this paper, we focus on 2D mesh network and use static XY routing for data transmission. Fig.1 shows 2D mesh network architecture.

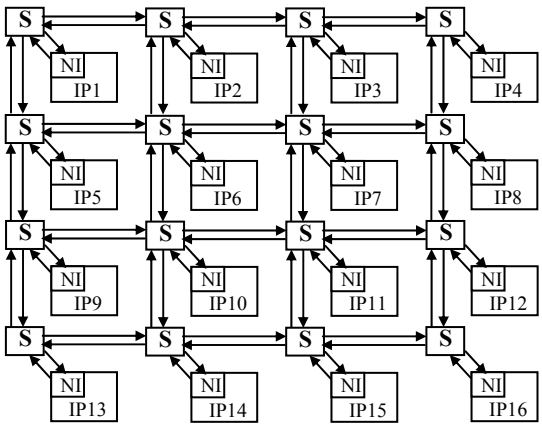


Fig. 1. 2D mesh NoC architecture

Manuscript received July 15, 2011; revised August 10, 2011. This work was supported in part by the National Natural Science Foundation of China (61076019), Jiangsu province scientific support plan (BE2010003) and Aeronautical Science Foundation of China (20115552031).

Ning Wu is with the College of Electrical and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 210016, China (e-mail: wunee@nuaa.edu.cn).

Yifeng Mu is with the College of Electrical and Information Engineering, NUAA, Nanjing 210016, China (e-mail: zealotmyf@163.com).

Fen Ge is with the College of Electrical and Information Engineering, NUAA, Nanjing 210016, China (e-mail: gefen@nuaa.edu.cn).

### A. Energy Model

Each switch connects four resource nodes and each resource node places one IP core.

We use the energy model presented in [8] by Hu et al:

$$E_{bit}^{i,j} = n_{switch} \times E_{Sbit} + (n_{switch} - 1) \times E_{Lbit} \quad (1)$$

where $E_{bit}^{i,j}$ represents the energy consumed by one bit of data transported from tile i to tile j, $E_{Sbit}$ the energy consumed by one bit of data transported a switch, $E_{Lbit}$ the energy consumed by one bit of data on the link between two switch and $n_{switch}$ the number of switch one bit of data encounters when it is transported from tile i to tile j. Note that, for the 2D mesh network with XY routing, (1) shows that the average energy consumption of sending one bit of data from tile i to tile j is determined by the Manhattan distance between these two tiles.

### B. Latency Model

The packet latency $T_{i,j}$ is defined as the amount of time from the generation of head flit in the source switch to the receiving of the packet's tail flit at the destination switch. To the wormhole route NoC,

$$T_{i,j} = (T_b + T_w) \times n_{switch} + T_b \times (B - 1) \quad (2)$$

where $T_b$ is the time that a flit transmits through a switch and one link in case of no conflict, $T_w$ the header flit waiting time at one switch in case of conflict, $B$ the flit number of a packet. $T_b$ and $B$ are constant, so the optimization of $T_{i,j}$ depends on $T_w$ and $n_{switch}$. $T_w$ is affected by network congestion. The most effective way to reduce $T_w$ is to balance the link load, so as to avoid network congestion. $n_{switch}$ is already considered in energy model.

### C. Objective Function

The optimization objectives are energy consumption and latency. The energy consumption (E(C)) is defined by (3).

$$E(C) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j} \times E_{bit}^{i,j} \quad (3)$$

where $w_{i,j}$ is the communication volume between tile i and j, N the number of tiles.

We optimize the latency through balancing the link load. The link load variance is defined by (4).

$$VAR(L) = \sum_{i=1}^{M} [Load(l_i) - Load(l)_{avg}]^2 / M \quad (4)$$

where $l_i$ is the $i^{th}$ link, $M$ the total number of links, $Load(l_i)$ the load of link $l_i$, and $Load(l)_{avg}$ the average load. The variance represents the discrete level of distribution, the greater variance the more uneven. So the optimization of latency is to minimize the link load variance.

The unified cost function can be represented by (5)

$$\cos t = \lambda \times E(c) + (1 - \lambda) \times VAR(L) \quad (5)$$

where λ is the ratio, which is used to adjust the proportion between communication energy consumption and latency in the cost function. Therefore, the objective function would be

$$\min(\cos t) \quad (6)$$

The scope of λ is 0~1. When λ=1, it means minimization of energy consumption. When λ=0, it means minimization of latency. When taking a value among 0~1, the optimization reflects a compromise between energy consumption and latency.

### D. Mapping Problem

To formulate the IP mapping problem in a more formal way, we need to introduce the following concepts:

Definition 1: An IP communication Characterization Graph (APCG) Ω=G(C,A) is a directed graph, where each vertex $c_i$ represents one IP, and each directed arc $a_{i,j}$ represents the communication from $c_i$ to $c_j$. The following quantities are associated with each $a_{i,j}$ as arc properties:

$v(a_{i,j})$: arc volume from vertex $c_i$ to $c_j$, which stands for the communication volume from $c_i$ to $c_j$.

$b(a_{i,j})$: arc bandwidth requirement from vertex $c_i$ to $c_j$, which stands for the minimum bandwidth that should be allocated by the communication network.

Definition 2: An Architecture Characterization Graph(ARCG) Π=G(T,P) is a directed graph, where each vertex $t_i$ represents one tile in the architecture, and each directed arc $p_{i,j}$ represents the routing path from $t_i$ to $t_j$. The following quantities are associated with each $p_{i,j}$ as arc properties:

$e(p_{i,j})$: arc cost from vertex from $t_i$ to $t_j$, which represents the average energy consumption of sending one bit of data from $t_i$ to $t_j$.

$L(p_{i,j})$: the set of links that make up the path $p_{i,j}$.

Using the above graph representations, and assuming map() as the mapping function from APCG to ARCG, the energy and link load can be calculated by (7) and (8).

$$E(C) = \sum_{\forall a_{i,j}} v(a_{i,j}) \times e(p_{map(c_i),map(c_j)})\} \quad (7)$$

$$Load(l_k) = \sum_{\forall a_{i,j}} v(a_{i,j}) \times f(l_k, p_{map(c_i),map(c_j)}) \quad (8)$$

where

$$f(l_k, p_{m,n}) = \begin{cases} 0 & l_k \notin L(p_{m,n}) \\ 1 & l_k \in L(p_{m,n}) \end{cases} \quad (9)$$

And the average load is calculated by (10).

$$Load(l)_{avg} = \sum_{i=1}^{M} Load(l_i) / M \quad (10)$$

We can get the cost function using (4), (5), (7), (8), (9) and (10).

The problem of minimizing the communication energy consumption can be formulated as: given an APCG and an ARCG that satisfy size(APCG) ⩽ size(ARCG), find a mapping function map() from APCG to ARCG which achieves (6),

such that:

$$\forall c_i \in C,\ map(c_i) \in T \quad (11)$$

$$\forall c_i \neq c_j \in C,\ map(c_i) \neq map(c_j) \quad (12)$$

$$B(l_k) \geq \sum_{\forall a_{i,j}} b(a_{i,j}) \times f(l_k, p_{map(c_i),map(c_j)}) \quad (13)$$

where $B(l_k)$ is the bandwidth of link $l_k$.

Conditions (11) and (12) mean that each IP should be mapped to one tile exactly and no tile can host more than one IP. Equation (13) guarantees that the load of any link will not exceed its bandwidth.

## III. IMPROVED MMAS

In MMAS, the elicitation information is obtained by using heuristics ($\eta$). The heuristics is calculated by (14).

$$\eta_{i,j} = \frac{f_i}{d_j} \tag{14}$$

where $f_i$ stands for the total communication volume between ith IP core and others, $d_i$ the total distance between ith IP and others. The heuristics shows the IP core with larger communication volume should be placed to the center of NoC architecture. But when we construct the solution for mapping problem, we map the IP core one by one. So the $f_i$ is a fixed value, and $\eta$ would indicate that each IP core should be mapped to the centre. Hence, the convergence rate of the algorithm gets slower, even we could find the best solution.

Consequently we obtain the elicitation information via priority mapping of IP core with larger communication volume instead of using heuristics.

## IV. GA-MMAS ALGORITHM

The basic idea of GA-MMAS Algorithm is that: in the beginning, it uses such kind of GA, which reserves the best ones, with the advantage of speediness and randomness. Then it initializes the pheromone for MMAS using the results of GA. Finally, it uses improved MMAS because of its precision and forward feedback to get the final results.

### A. GA in GA-MMAS

We first use GA to get some better results. The key steps of our GA are as follows:

1) Generate a random population of valid solutions. Here each individual member of the population is a chromosome, which represents a concrete design plan of vertex mapping on IPs. A chromosome is constructed as follows:

(a) Every chromosome is a series of genes.

(b) Every gene represents an IP core. Its concrete value is the vertex number in ARCG. Every chromosome, consisting of genes, should satisfy (13). Fig.2 shows one chromosome.

| 5 | 1 | 3 | 4 | 0 | 2 | 8 | 7 | 6 |
|---|---|---|---|---|---|---|---|---|

Fig. 2. Chrome code

The lenth of the chromosome is 9, so there are 9 IP cores in APCG. The $0^{th}$ number '5' represents the $0^{th}$ IP core is mapped to vertex 5.

The number of individuals in one generation is 100.

2) Calculate the fitness function of every member as the reciprocal of its system communication energy.

$$fitness = \frac{1}{\cos t} \tag{15}$$

3) Generate a better population from the old population. We use improved order-crossover method and mutation method to generate the next new population and ensure that the best individual of current population is saved for the next new population. Fig.3 shows the improved order-crossover method.

A and B are two old chromosomes and A* and B* are two new ones.

The probability of crossover is 0.95 and the one of mutation is 0.05.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A | 5 | 1 | 3 | 4 | 0 | 2 | 8 | 7 | 6 |
| B | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A* | 5 | 6 | 7 | 8 | 1 | 3 | 4 | 0 | 2 |
| B* | 2 | 8 | 7 | 6 | 0 | 1 | 3 | 4 | 5 |

Fig. 3. Improved order-crossover

4) GA stops when the number of generation reaches at 1000.

### B. initializing the pheromone

We chose the best x×y/2 individual members in the population of last generation to form matrix T1. Form another matrix T2 by individual members generated randomly. Then form matrix P using T1 and T2 [9].

Assume that $p_{i,j}$ in the matrix $\boldsymbol{P}$ represents the mapping times of $i^{th}$ IP to $j^{th}$ node. So the element $\tau_{i,j}$ of pheromone matrix $\boldsymbol{\tau}$, which represents the pheromone of $i^{th}$ IP mapping to $j^{th}$ node, is $p_{i,j}/(x \times y)$. For example, assume the architecture of NoC is 2D mesh of 2×2.The best two individuals in the last generation of GA are 2013 and 2031. 1302 and 1230 are generated randomly, so $T1 = \begin{bmatrix} 2 & 0 & 1 & 3 \\ 2 & 0 & 3 & 1 \end{bmatrix}$ and $T2 = \begin{bmatrix} 1 & 3 & 0 & 2 \\ 1 & 2 & 3 & 0 \end{bmatrix}$.

Then we can get $P$ and $\tau$, that is:

$$P = \begin{bmatrix} 2 & 0 & 1 & 3 \\ 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 1 & 2 & 3 & 0 \end{bmatrix}, \quad \tau = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0 & 0.5 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}.$$

### C. MMAS in GA-MMAS

Finally, we use MMAS to get the final results.

In our MMAS, each ant in the ant system has a path-table which represents a concrete design plan of vertex mapping on IPs. The location of path-table represents the number of IP and the value is vertex number in ARCG. Each ant also has a state-table which notes the unmapped nodes. The number of ants is the number of IP cores.

The key steps of our MMAS are as follows:

1) Calculate the total communication volume of each IP from or to other IPs.

2) Assume Q is a queue which is made of IPs order by total communication volume of IP.

3) Pick the IP orderly from Q and map to a node according to the rules as follows.

Assume q is a random number between 0 and 1, $q_0$ a constant between 0 and 1. If $q < q_0$, the mapping node is given by (16)

$$j = \arg\max_{s \in allowed_k} (\tau_{is}^{\alpha}) \tag{16}$$

where $\alpha$, a parameter, determines the relative importance of pheromone which is set to 0.8 and $allowed_k$ is the state-table of Ant k.

Otherwise if $q \geq q_0$, assume r is a random between 0 and 1. Then compute the probability given by (17) with which $i^{th}$ IP

is mapped to $j^{th}$ node and accumulate the probability to s until s>r. Then the corresponding node is the mapping node.

$$p_{i,j}^{k}(t)=\begin{cases}\dfrac{(\tau_{ij}(t))^{\alpha}}{\sum_{r\in allowed}(\tau_{ir}(t))^{\alpha}} & j\in allowed\\ 0 & j\notin allowed\end{cases} \quad (17)$$

4) Put the node into path-table and delete it from the state-table when the node is selected for an IP core.

5) Do step (3) and (4) until all IPs are mapped. Then one mapping result is gotten.

6) Do step (3), (4) and (5) for all ants to complete one cycle and each ant is one result.

7) Calculate the fitness function of every ant using (15). Then use 2-Opt method for local search for the best ant.

8) Update pheromone just for the best ant when one cycle completes using (18) where ρ is a pheromone decay parameter which is set to 0.8 and (19) where fitness(best solution) is the fitness of the best ant.

$$\tau_{i,j}(t+1)=\rho\times\tau_{i,j}(t)+\Delta\tau \quad (18)$$

$$\Delta\tau=fitness(bestsolution) \quad (19)$$

To avoid search stagnation, the pheromone is increased proportionally to the difference between $\tau_{min}$ and $\tau_{max}$ given by (20) and (21).

$$\tau_{\max}=\frac{1}{1-\rho}fitness(bestsolution) \quad (20)$$

$$\tau_{\min}=\frac{1}{5}\tau_{\max} \quad (21)$$

9) MMAS stops when the number of cycle reaches at 1000. The best result is the best one in last cycle.

## V. Experimental Results

We have carried out various experiments to verify the efficiency of our algorithm. We use GA, ACA and MMAS for comparison. All of the algorithms are complied by C++ and run on Windows XP in a computer with Intel P IV 3.4 GHz CPU and 1 GB memory.

### *A. Random benchmark experiments*

All tasks are generated by TGFF [10].Table I shows the detail of each task, including number of IP cores, number of directed arcs, total communication volume and NoC Size.

TABLE I
Experiments Detail

| Task-No | IP-No | directed arc-No | Total volume of communication | NoC Size |
|---|---|---|---|---|
| 1 | 10 | 12 | 1634 | 4 × 3 |
| 2 | 12 | 12 | 3466 | 4 × 3 |
| 3 | 16 | 33 | 609257 | 4 × 4 |
| 4 | 20 | 24 | 2640 | 5 × 4 |
| 5 | 20 | 24 | 2640 | 5 × 5 |
| 6 | 20 | 37 | 4176 | 5 × 4 |
| 7 | 20 | 37 | 4176 | 5 × 5 |
| 8 | 25 | 33 | 684271 | 5 × 5 |
| 9 | 40 | 72 | 80951 | 7 × 6 |
| 10 | 48 | 95 | 98840 | 7 × 7 |

We set the cost of random mapping to 1. Fig.4, Fig.5 and Fig.6 shows the cost comparative results.

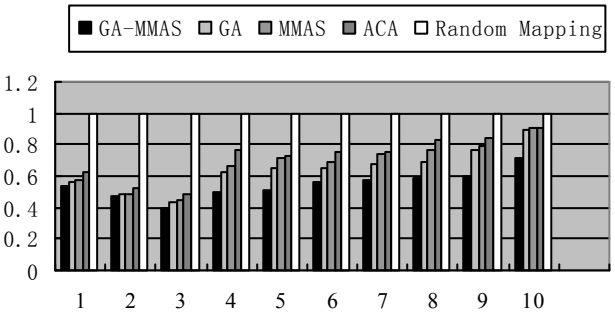


Fig. 4. cost comparative results (λ=1)

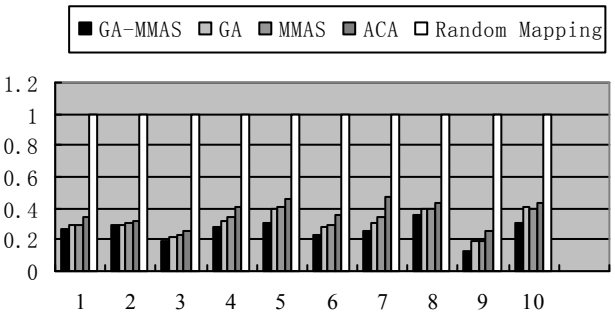


Fig. 5. cost comparative results (λ=0.5)

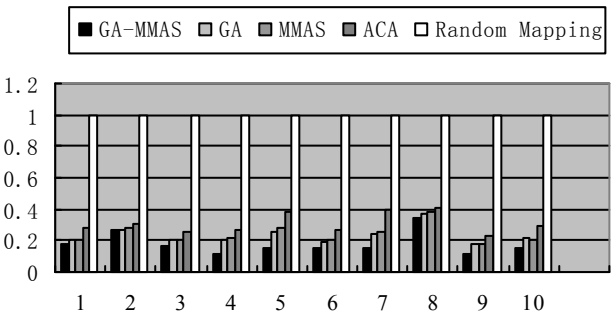


Fig. 6. cost comparative results (λ=0)

Experimental results show that the cost is saved when we use those kinds of algorithm for different tasks. We calculate the cost saving results (%) by GA-MMAS compared to random mapping, GA, ACA and MMAS from the figures above. Table II shows the results.

TABLE II
Time Comparative Results

| Algorithm | λ=1 | | λ=0.5 | | λ=0 | |
|---|---|---|---|---|---|---|
| | best/ % | avera ge/% | best/ % | avera ge/% | best/ % | avera ge/% |
| Random mapping | 60 | 45 | 87 | 71 | 89 | 80 |
| GA | 25 | 20 | 33 | 20 | 42 | 25 |
| ACA | 30 | 20 | 49 | 30 | 61 | 45 |
| MMAS | 30 | 20 | 35 | 20 | 43 | 30 |

As shown in Table II, compared to the other algorithm, the algorithm can save the cost most whenever λ=1, λ=0.5 or λ=0. This is the result of that the algorithm combines MMAS with GA to make compensation to their disadvantages by their advantages.

We also have compared the computer time of each algorithm. Table III shows the comparative result.

TABLE III
TIME COMPARATIVE RESULTS

| Task-No | GA | ACA | MMAS | GA-MMAS |
|---|---|---|---|---|
| 1 | 1.8s | 1.8s | 8.5s | 10s |
| 2 | 2s | 2s | 9s | 11s |
| 3 | 3s | 5.5s | 10s | 12s |
| 4 | 3s | 5s | 13s | 14s |
| 5 | 5s | 7s | 20s | 21s |
| 6 | 7s | 10s | 25s | 27s |
| 7 | 6s | 8s | 21s | 23s |
| 8 | 7s | 12s | 30s | 31s |
| 9 | 18s | 72s | 198s | 168s |
| 10 | 24s | 132s | 368s | 321s |

Experimental results show that the computing time of GA-MMAS algorithm is more than the one of GA and ACA, but is similar to the one of MMAS. With the number of IP increasing, the compute time of GA-MMAS is growing much more. Then we carried out one experiment with a task of 96 IPs. The time is about 20 minutes and could be accepted.

### B. A Video/Audio Application

In this paper, we carried one experiment for a multi-media system (MMS) which is consisted by H.263 video encoder/decoder and MP3 audio encoder/decoder. The multi-media system can be divided into 25 IP cores. Fig.7 shows the communication volume between the IP cores.

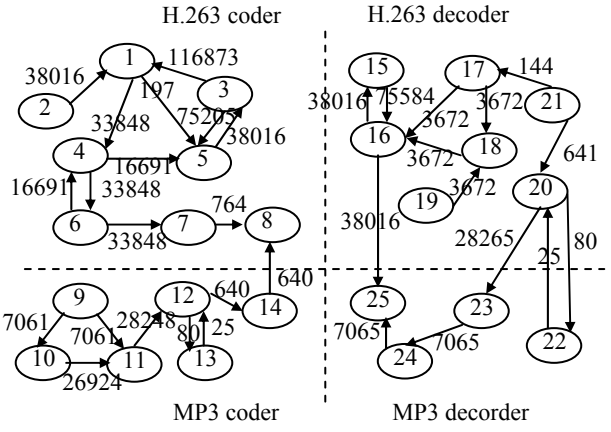


Fig. 7. IP communication graph for MMS

In 0.18μm technology, assume that the link length is 2mm, the total line capacitance is 0.5fF/μm and voltage swing is 3.3v, then $E_{Lbit}$ is 5.445pJ and $E_{Sbit}$ is 0.43pJ [11]. Based on the above parameters, the cost is calculated by GA-MMAS and other algorithms respectively as shown in Table IV.

TABLE IV
ENERGY COMPARATIVE RESULT FOR MMS

| Algorithm | λ=1 | | λ=0.5 | | λ=0 | |
|---|---|---|---|---|---|---|
| | cost/ $10^6$ | ratio | cost/ $10^6$ | ratio | cost/ $10^6$ | ratio |
| GA-MMAS | 35.3 | 1 | 204.8 | 1 | 371.1 | 1 |
| Random mapping | 59.6 | 1.69 | 570.3 | 2.79 | 1080.9 | 2.91 |
| GA | 41.0 | 1.15 | 220.1 | 1.07 | 399.2 | 1.08 |
| ACA | 49.5 | 1.39 | 247.5 | 1.21 | 439.0 | 1.18 |
| MMAS | 45.6 | 1.29 | 226.5 | 1.11 | 413.1 | 1.11 |

In this experiment, the computing time of GA-MMAS is 40s. When λ=1, GA-MMAS can save about 41%, 14%, 29% and 23% of cost compared to random mapping, GA, ACA and MMAS respectively. When λ=0.5, the algorithm can save about 64%, 7%, 17% and 10%. When λ=0, the saving results are 66%, 7%, 15% and 10%.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we optimized latency indirectly by optimizing the distribution of link load, and presented a GA-MMAS algorithm aiming at the problem of energy- and latency-aware IP mapping for Network-on-Chip with the 2Dmesh architecture. This algorithm use GA to initialize the pheromone for MMAS, making compensation to the lack of pheromone in the early stage of MMAS. The heuristics is not used in the computation of probability. Instead of it, a new method is used for better results in which the IP core with lager communication volume is mapped firstly. Several experiments are carried out to verify the efficiency of the algorithm. Experimental results show that the algorithm saves about 36%~60%, 3%~25%, 10%~30% and 3%~30% compared to random mapping, GA, ACA and MMAS when λ=1; when λ=0.5, the results are 63%~85%, 1%~33%, 10%~44% and 3%~30%; and 65%~87%, 1%~39%, 11%~58% and 4%~35% when λ=0.

Currently, 3D integrated circuits (ICs), with the benefits of short interconnect length, high density package and low power, have the potential for enhancing system performance. The synergy between the 3D ICs and NoC making it possible to scale NoC over the third dimension and resulting in 3D NoC, which is now a hot topic today. Therefore, our future work is to adapt the proposed algorithm to 3D NoC architecture, for optimizing energy consumption and latency.

## REFERENCES

[1] OGRAS U Y, HU Jingcao and MARCHLESCU R. "Key research problems in NoC design: a holistic perspective," in *Proc. 3rd IEEE Int. Conf. Hardware/Software Codesign and System Synthesis*, Jersey City, 2005,pp. 69–74.

[2] Morgan.A.A, Elmiligi.H, El-kharashi.M.W and Gebali.F, "Multi-objective optimization of NoC standard architectures using Genetic Algorithms," in *Proc. IEEE Int. Symposium. Signal Processing and Information Technology,* Luxor, Egypt, 2010, pp: 85-90

[3] Zhou W B, Zhang Y and Mao Z G. "An application specific NoC mapping for optimized delay," in *Proc. IEEE Int. Conf. DTIS*, 2006, pp. 184–188

[4] Gen fen and Wu ning. "Co-Mapping Algorithm for Network on Chip with Optimal Power Consumption," *Journal of Applied Sciences,* vol. 26, no. 6, 2008, pp.606–216.

[5] Yanhua Liu, Ying Ruan, Zongsheng Lai and Weiping Jing, "Energy and Thermal Aware Mapping for Mesh-based NoC Architectures Using Multi-objective Ant Colony Algorithm," in *Proc. 3rd International Conf. Computer Research and Development*, 2011, pp.407–411

[6] Ganmin Zhou, Yongsheng Yin, Yonghua Hu and Minglun Gao. "NoC mapping based on ant colony optimization algorithm," *Journal of Computer Engineering and Applications*, no. 18, pp. 7–10, 2005

[7] Ning Wu, Yifeng Mu, Fang Zhou and Fen Ge. GA-MMAS: an Energy-aware Mapping Algorithm for 2D Network-on-Chip. Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2011, WCECS 2011,19-21 October, 2011, San Francisco, USA, pp: 739-743

[8] HU Jingcao, MARCHLESCU R. "Energy-aware mapping for tile-based NoC architectures under performance constraints," *in Proc.*

*Asia South Pacific Conf. Design Automation,* Kitakyushu, 2003, pp. 233–239.

[9] Wu Jiao-feng. "Research on improved performance of Ant Colony Algorithm by Genetic Algorithm," MA.Eng. dissertation, Taiyuan University of Technology, 2007

[10] Dick R P, Rhodes D L and Wolf W. "TGFF: task graphs for free," in *Proc. 6th International Workshop on Hardware/Software Codesign*, Seattle, Wahsington, 1998, pp. 97–101.

[11] Ye T T, Benini L and Micheli G De. "Analysis of power consumption on switch fabrics in network in routers," *in Proc. Design Automation Conf.*, 2002, pp. 524–529