Power-aware Meta-heuristic Core Mapping Approaches for Network on Chips

Mehdi Taassori, Sener Uysal
Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Famagusta, Mersin 10, Turkey

Abstract— Network on Chip (NoC) has been introduced to support communications demand in System on Chip (SoC). Power consumption is a controversial issue in on-chip interconnections. Due to this issue and limited resources in NoCs such as wires, switches and virtual channels, mapping problem which are dealing with obtaining an appropriate position in topology, plays crucial role in design of NoCs. In this paper, we utilize Genetic Algorithm (GA) and Simulated Annealing (SA) as meta-heuristic algorithms to solve the Quadratic Assignment Problem (QAP) and map the tasks to the cores on mesh-based NoCs. Experimental results reveal that meta-heuristic algorithms not only reduce the power dissipation but also improve the performance in NoCs.

Index Terms— Power consumption; Mapping; Meta-heuristic; Genetic Algorithm; Simulated Annealing; Network on Chip.

1 Introduction

As technology shirinked, designers use many intellectual property (IP) on a System on Chip (SoC). With this trend the communication in SoC has a great impact on power consumption and transmission delay in on-chip interconnection [1]. Network on Chip (NoC) has been recommended as a solution for SoC's communication problems [2].

With the advances in technology, the power dissipation has become a major issue [1] in NoCs which is composed of link and router's power.

Due to the importance of power dissipation and limited resources in NoCs such as wires, switches and virtual channels, mapping problem tackles with finding an appropriate position in topology, plays crucial role to design an optimal layout for NoCs [3].

In [4], Quadratic Assignment Problem (QAP) which is classified as NP-hard problem is used to solve the layout problem.

Researchers in [3] addressed an analythical optimization approach to map the tasks to the cores on NoCs. In this paper, we use Genetic Algorithm (GA) and Simulated Annealing (SA) as meta-heuristic methods to solve QAP and map the tasks to the cores on the mesh based NoCs. The goal of this research is not only to reduce the power consumption but also to improve the performance in NoCs.

2 Literature Review

Analytical and meta-heuristic algorithms are introduced to solve NP-hard problems [4-8]. Researchers in [3], [9-16], proposed different analytical and meta- heuristic approaches to minimize the power consumption in NoCs.

The authors in [3] proposed a mathematical optimization method to map the tasks to the cores in NoCs under link length and bandwidth constraints. They presented a linearized form of QAP to solve the layout problem. Their objective was power reduction that can be achieved through mapping along with finding the optimum number of routers and virtual channels in NoCs. In [15], a genetic algorithm with the goal of power and router reduction in NoC architectures is presented. Another heuristic method for mapping the tasks to the cores on mesh-based NoC is introduced in [11] by minimizing the communication delay under bandwidth constraint.

Different mapping algorithms for mesh-based NoC are presented in [11], [15], [17-20].

3 Mathematical Model

To solve the mapping problem in NoCs a mathematical approach as Quadratic Assignment Problem (QAP) is presented [3]. QAP, which is used in engineering sciences [21, 22], is a useful analytical optimization algorithm. It is categorized as NP-hard problem [21], [23] and consequently non-polynomial CPU running time is very high, therefore, researchers use meta-heuristic algorithms [23, 24]. We used QAP to map the tasks to the cores in NoCs.
The parameters and variables in the mathematical model are described as follows:

n: The number of tasks and cores of NoC.

i,j: Indexes of the tasks.

k,l: Indexes of the cores.

X_{ik}: Binary variable which is 1 if task i is mapped to core k, otherwise, it is zero.

f_{ij}: Bandwidth between tasks i and j.

d_{kl}: Distance between cores k and l.

- *Mehdi Taassori, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Famagusta, Mersin 10, Turkey. E-mail:mehdi.taassori@emu.edu.tr*
- *Sener Uysal, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Famagusta, Mersin 10, Turkey. E-mail:sener.uysal@emu.edu.tr*

P)

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} X_{ik} X_{jl} \qquad (1)$$

subject to

$$\sum_{i=1}^{n} X_{ik} = 1 \qquad \forall k \in \{1,2,\ldots,n\} \qquad (2)$$

$$\sum_{k=1}^{n} X_{ik} = 1 \qquad \forall i \in \{1,2,\ldots,n\} \qquad (3)$$

$$X_{ik} \in \{0,1\} \qquad \forall i,k \in \{1,2,\ldots,n\} \qquad (4)$$

The minimization objective function (1) evaluates summation of distances for all pairs of cores which is weighted by their bandwidth. The constraint (2) emphasises that only 1 task can be mapped to each core and constraint (3) emphasises that only 1 core can be mapped to each task. The constraint (4) defines binary nature of the X variables.

3.1 Genetic Algorithm (GA)

QAP is a combinatorial optimization problem and GA is a population based evolutionary algorithm which is used to solve combinatorial optimization problems [5], [25-29]. GA to solve the QAP starts with a population that is generated randomlly (chromosomes). Any solution should be evaluated by the objective function (1). Then, percentage of the generated population (P_r%), that includes the worst solution, is randomly selected to regenerate and one or some of the remaining solutions are selected as parents. Parents using crossover and mutation operators, generate new solutions by the name of child or offspring. The objective function (1) assesses the new solution and if the objective function value is better than the worst solution then the worst solution is replaced with the new one. Regeneration, selection mechanism, offspring generation, eveluation process and replacement all together are one iteration in GA. After limited number of iterations the GA is terminated. In the final population, the best solution is selected as a solution of the GA.

3.1.1 Population of GA

Population of GA that is an array of integer numbers, is depicted in Figure 1. As shown in Figure 1, the value and index of element of array illustrate the number of tasks and cores, respectively. As an example, task 3 is mapped to the core 7. This is an example for VOPD benchmark; hence, 12 tasks are mapped to the 12 cores.

Core	1	2	3	4	5	6	7	8	9	10	11	12
Task	4	9	2	7	10	1	3	6	11	5	12	8

Fig. 1. Solution representation of GA.

3.1.2 Neighbor generation for GA

A percentage of the population (Pr%) is regenerated. Thus, a new solution is produced instead of each solution. Then from the remaining solutions the offspring solution is generated by applying the Partial Mapped Crossover [30, 31] and swap operator. Firstly, a pair of parents from the *(1-Pr)*% of the population is selected by the selection mechanism of the parents. Secondly, the Partial Mapped Crossover is applied on the selected parents to find an offspring. Finally, swap operator is used on offspring to generate final offspring.

3.1.3 Selection mechanism of parents

The fitness value which is defined as *1/objective function* is assigned to each solution of the population in each iteration of GA. Due to the minimization type of the objective function (1), the better solution has greater fitness value. The selection mechanism of the parents is Roulette–Wheel selection [5], [25, 26]. In this approach the better solution is a candidate to be selected as parent.

3.2 Simulated Annealing (SA)

Simulated annealing that is introduced by [32] as a metaheuristic optimization algorithm is able to solve combinatorial optimization problems. SA has been suggested to solve layout problem [33-37].

SA is an iterative improving algorithm to produce a solution randomly. The generation process is used to generate a new solution namely neighbor solution. In general, neighbor solution is founded by swap operator.

SA algorithm uses the following parameters:

- T_0: Initial temperature
- r: Cooling ratio such that $0 \prec r \prec 1$
- T_f : Final temperature
- N_{it} : Number of iterations
- x`: Neighbor solution
- x : Initial solution

To solve the QAP, SA considers an initial solution (x) and temperature (T_0). Under this condition a new solution that is the neighbor solution is generated (x`). ($x$`) is evaluated according to the objective function (1).

The same temperature is used for this neighbor solution and the solution is evaluated for a limited number of iterations (N_{it}). In the next step temperature is cooled down based on the cooling ratio (r) such that $0 \prec r \prec 1$ and in the new temperature the same process is done. This algorithm continues untill the final temperature (T_f).

3.2.1 Solution representation in SA

Solution representation in SA is an array of integer numbers as shown in Figure 1.

3.2.2 Neighbor generation mechanism of SA

Neighbor solution in SA is generated by a double swap operator. In this operator two integer random number form the interval [1,n] is generated where n is the number of tasks or cores. In the array of the solution, the numbers of the tasks that are generated randomly are swapped. The obtained array is the neighbor solution.

4 Experimental Results

We compared meta-heuristic algorithms with five benchmarks; H.263 video encoder, H.263 video decoder, MP3 audio encoder , Video object plane decoder (VOPD) and Multi-window display (MWD) [14].
Table1 shows the benchmark's characteristics. In the third and forth column the number of tasks and cores are shown, repectively.

TABLE 1
Benchmarks' Characteristics

Graph	Graph ID	Task	Core
H.263 encoder	G1	8	11
MWD	G2	12	13
VOPD	G3	12	15
H.263 enc MP3 enc	G4	14	19
H.263 enc H.263 dec	G5	15	19

The meta-heuristic algorithms were coded in Matlab and run on a PC with an Intel Core 2 Duo 2.53 GHz processor and 4.00 GB RAM.

The parameters level of GA is considered as follows:

- Number of population: 50, 150, 300.
- P_r: 50, 65, 80.
- Number of iterations: 200, 400, 600.

GA with 27 combinations which are taken from the above parameters level is run. Then among the solutions the best one is selected as the best layout.

The parameters level of SA is considered as follows:

- Initial temperature: 50, 150, 300.
- Cooling ratio: 0.85, 0.95, 0.99.
- Number of iterations: 50, 250, 400.
- Final temperature: 5.

SA with 27 combinations which are taken from the above parameters level is run. Then among the solutions the best one is selected as the best layout.

The power consumption of NoC is summation of link power and router's power dissipation [10]. Link power includes self and coupling capacitances that is calculated by $P_{link}=aCV^2{}_{dd}f$, where a is the switching activity, C is self and coupling capacitance, the voltage of the power supply is V_{dd} and f is the clock frequency. The benchmarks are implemented in 65 nm technology. According to the critical path of the system, frequency is 500 MHz. Regarding to the International Technology Roadmap for Semiconductors [1], V_{dd} is 1 Volt. The length of the wires considered as 2 mm while the self and coupling capacitance of the wire links considered as 0.2 pF/mm and 0.6 pF/mm, respectively. Transitions of the wires are counted by Modelsim and power of the routers is evaluated with power compiler from Synopsys.

The results of the power consumption for GA and SA compared to the non-optimized NoC are depicted in Figure 2. In this Figure benchmarks are indicated in horizontal axis and the normalized value of power dissipation is indicated in the vertical axis. The vertical axis in Figure 2 is normalized to the corresponding power consumption by the non-optimized layout.

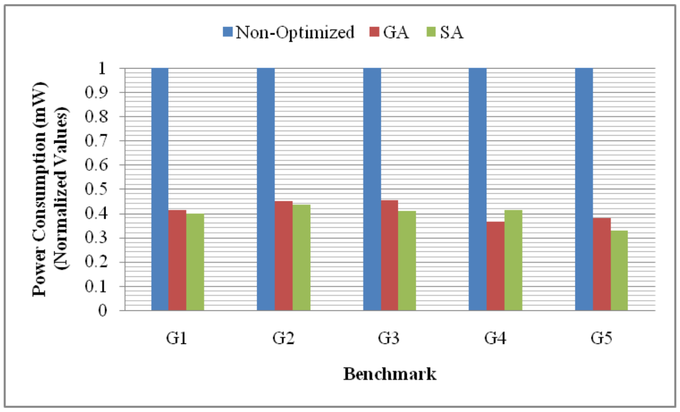

Fig. 2. Power comparison of non-optimized, GA and SA.

On an average GA and SA consumes 58.7%, and 60.2% lower power dissipation compared to the non-optimized NoC, respectively.
Latency is compared in Figure 3. Benchmarks are indicated in the horizontal axis and normalized value of latency is demonstrated in the vertical axis which is normalized to the corresponding latency by the non-optimized layout.

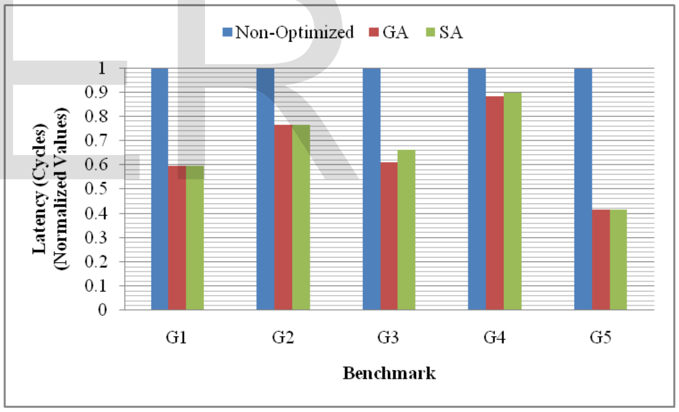

Fig. 3. Latency comparison of non-optimized, GA and SA.

As shown in Figure 3, on an average GA and SA improve the latency 34.5%, and 33.1% compared to the non-optimized NoC, respectively.

5 Conclusion

In this paper, meta-heuristic algorithms such as GA and SA are used to solve QAP and map the tasks to the cores on mesh-based NoCs. Experimental results show that not only the power consumption is decreased but also the performance of the NoC is improved.

References

[1] International Technology Roadmap for Semiconductors (ITRS), http://www.itrs.net, 2010.

[2] L. Benini and G. De Micheli, "Networks on chips: A New SoCParadigm," Computer, vol. 35, no. 1, pp. 70–78, 2002.

[3] M. Taassori, M. Taassori, S. Niroomand, B. Vizvari, S. Uysal, A. Hadi-Vencheh, "OPAIC: An optimization technique to improve energy consumption and performance in application specific network on chips, " Measurement, vol. 74, pp. 208-220, 2015.

[4] S. Niroomand, S. Takács, and B. Vizvári, "To lay out or not to lay out?," Ann. Oper. Res., vol. 191, no. 1, pp. 183–192, 2011.

[5] A. Mahmoodi-Rad, S. Molla-Alizadeh-Zavardehi, R. Dehghan, M. Sanei, and S. Niroomand, "Genetic and differential evolution algorithms for the allocation of customers to potential distribution centers in a fuzzy environment," Int. J. Adv. Manuf. Technol., vol. 70, no. 9–12, pp. 1939–1954, 2014.

[6] S. Niroomand and B. Vizvari, "Exact mathematical formulations and metaheuristic algorithms for production cost minimization: a case study of the cable industry," Int. Trans. Oper. Res., vol. 22, no. 3, pp. 519–544, 2015.

[7] M. Hajiaghaei-Keshteli, S. M. Sajadifar, and R. Haji, "Determination of the economical policy of a three-echelon inventory system with (R, Q) ordering policy and information sharing," Int. J. Adv. Manuf. Technol., vol. 55, no. 5–8, pp. 831–841, 2011.

[8] M. Hajiaghaei-Keshteli and M. Aminnayeri, "Solving the integrated scheduling of production and rail transportation problem by Keshtel algorithm," Appl. Soft Comput., vol. 25, pp. 184–203, 2014.

[9] O. He, S. Dong, W. Jang, J. Bian, and D. Z. Pan, "UNISM: Unified scheduling and mapping for general networks on chip," IEEE Trans. Very Large Scale Integr. Syst., vol. 20, no. 8, pp. 1496–1509, 2012.

[10] K. Srinivasan, K. S. Chatha, and G. Konjevod, "Linear-programming-based techniques for synthesis of network-on-chip architectures," IEEE Trans. Very Large Scale Integr. Syst., vol. 14, no. 4, pp. 407–420, 2006.

[11] S. Murali and G. De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures," Des. Autom. Test Eur., Date, pp. 896–901, 2004.

[12] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," IEEE Trans. Comput. Des. Integr. Circuits Syst., vol. 24, no. 4, pp. 551–562, 2005.

[13] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," ASP-DAC, pp. 233–239, 2003.

[14] D. Bertozzi, A. Jalabert, S. Murali, and S. Member, "NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip," IEEE Trans. Parallel Distrib. Syst., vol. 16, no. 2, pp. 113–129, 2005.

[15] G. Leary, K. Srinivasan, K. Mehta, and K. S. Chatha, "Design of network-on-chip architectures with a genetic algorithm-based technique," IEEE Trans. Very Large Scale Integr. Syst., vol. 17, no. 5, pp. 674–687, 2009.

[16] K. S. Chatha, K. Srinivasan, G. Konjevod, "Automated Techniques for Synthesis of Application-Specific Network-on-Chip Architectures," IEEE Trans. Comput. Des. Integr. Circuits Syst., vol. 27, no. 8, pp. 1425–1438, 2008.

[17] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli, "A Methodology for Mapping Multiple Use-Cases onto Networks on Chips," Des. Autom. Test Eur., Date, vol. 1, pp. 1–6, 2006.

[18] J. Hu and R. Marculescu, "Exploiting the routing flexibility for energy/performance-aware mapping of regular NoC architectures," Des. Autom. Test Eur., Date, pp. 688-693, 2003.

[19] K. Srinivasan and K. S. Chatha, "A technique for low energy mapping and routing in network-on-chip architectures," ISLPED '05, pp. 387–392, 2005.

[20] P. K. Sahu, T. Shah, K. Manna, and S. Chattopadhyay, "Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization," IEEE Trans. Very Large Scale Integr. Syst., vol. 22, no. 2, pp. 300–312, 2014.

[21] A. F. Alkaya and E. Duman, "Combining and solving sequence dependent traveling salesman and quadratic assignment problems in PCB assembly," Discret. Appl. Math., vol. 192, no. 2015, pp. 2–16, 2015.

[22] M. Laurent and M. Seminaroti, "The quadratic assignment problem is easy for Robinsonian matrices with Toeplitz structure," Oper. Res. Lett., vol. 43, no. 1, pp. 103–109, 2015.

[23] B. Rostami and F. Malucelli, "A revised reformulation-linearization technique for the quadratic assignment problem," Discret. Optim., vol. 14, pp. 97–103, 2014.

[24] W. Adams and L. Waddell, "Linear programming insights into solvable cases of the quadratic assignment problem," Discret. Optim., vol. 14, pp. 46–60, 2014.

[25] M. Gen, R. Cheng, Genetic algorithms and engineering design, Wiley, New York, 1997.

[26] M. Gen, R. Cheng, Genetic algorithms and engineering optimization, Wiley, New York, 2000.

[27] M. Hajiaghaei-Keshteli, "The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches," Appl. Soft Comput. J., vol. 11, no. 2, pp. 2069–2078, 2011.

[28] S. Molla-Alizadeh-Zavardehi, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam, "Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prufer number representation," Expert Syst. Appl., vol. 38, no. 8, pp. 10462–10474, 2011.

[29] M. R. Delavar, M. Hajiaghaei-Keshteli, and S. Molla-Alizadeh-Zavardehi, "Genetic algorithms for coordinated scheduling of production and air transportation," Expert Syst. Appl., vol. 37, no. 12, pp. 8255–8266, 2010.

[30] L. Garcia-Hernandez, H. Pierreval, L. Salas-Morera, and a. Arauzo-Azofra, "Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm," Appl. Soft Comput. J., vol. 13, no. 4, pp. 1718–1727, 2013.

[31] A. Eiben, J. Smith, Introduction to evolutionary computing, Springer-Verlag, 2003.

[32] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, "Optimization by Simulated Annealing," Science, vol. 220, no. 4598, pp. 671–680, 1983.

[33] A. Baykasoğlu and N. N. Z. Gindy, "A simulated annealing algorithm for dynamic layout problem," Comput. Oper. Res., vol. 28, no. 14, pp. 1403–1426, 2001.

[34] R. Şahin, K. Ertoral, and O. Türkbey, "A simulated annealing heuristic for the dynamic layout problem with budget constraint," Comput. Ind. Eng., vol. 59, no. 2, pp. 308–313, 2010.

[35] S. Niroomand, A. Hadi-Vencheh, R. Şahin, and B. Vizvari, "Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems," Expert Syst. Appl., vol. 42, pp. 6586–6597, 2015.

[36] S. S. Heragu and Attahiru Sule Alfa, "Experimental analysis of simulated annealing based algorithms for the layout problem," Eur. J. Oper. Res., vol. 57, no. 2, pp. 190–202, 1992.

[37] M. Braglia, "Optimisation of a Simulated-Annealing-based Heuristic for Single Row Machine Layout Problem by Genetic Algorithm," Int. Trans. Oper. Res., vol. 3, no. 1, pp. 37–49, 1996.